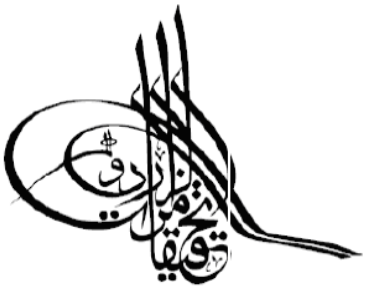

WORD SEGMENTATION FOR URDU OCR SYSTEM

MS Thesis

Submitted in Partial Fulfillment

Of the Requirements of the

Degree of

Master of Science (Computer Science)

AT

NATIONAL UNIVERSITY OF COMPUTER & EMERGING SCIENCES

LAHORE, PAKISTAN

DEPARTMENT OF COMPUTER SCIENCE

By

Misbah Akram

07L-0811

Approved:

_____

Head

(Department of Computer Science)

Approved by Committee Members:

Advisor

_____

Dr. Sarmad Hussain

Professor

FAST - National University

Other Members:

_____

Mr. Shafiq-ur-Rahman

Associate Professor

FAST - National University

The undersigned hereby certify that they have read and recommend to the Faculty of Graduate Studies for acceptance of a thesis entitled “Word Segmentation for Urdu OCR System" by Misbah Akram in partial fulfillment of the requirements for the degree of Master of Science.

Dated: August 2009

CONTENTS

FIGURES

TABLES

1. INTRODUCTION

Urdu language is a derivation of Indo-Aryan family of languages and more of its vocabulary is borrowed from the Hindi, Persian and Arabic languages. Urdu is the national language of Pakistan and it is spoken by 104 millions of speakers from all over the world. Urdu text is written using Arabic script and Perso-Arabic Nastalique style is mostly used for Urdu orthography [29][30].Urdu character set consists of 58 letters [1] which include characters from the Arabic and Persian character sets. It further expands its character set to represents sounds which are present in Urdu but not in Arabic or Persian. Urdu Character set is given in Figure 1-1 [1] (other sources may give slightly different set).

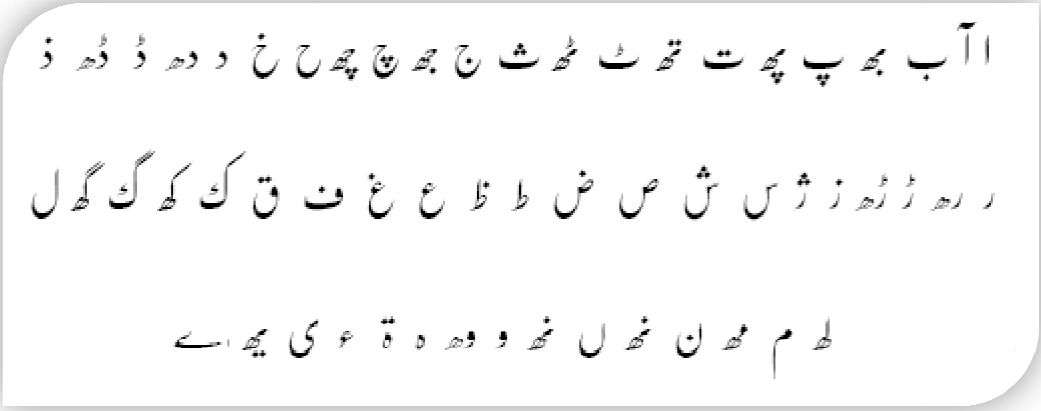

FIGURE 1-1 : URDU CHARACTER SET [1]

1.1. LIGATION AND CONTEXT SENSITIVE GLYPH SHAPING IN URDU TEXT

Urdu text script is cursive in nature means in this script letters are joined together into units to form words. These connected units are called ligatures. Urdu character set is composed of two kinds of characters, joiners and non-joiners. These two groups are also called separators and non-

separators respectively. Figure 1-2 shows the list of the separators or non-joiner from the character set given in figure 1-1.

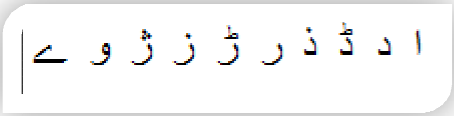

FIGURE 1-2 : SEPERATORS / NON- JOINERS IN URDU TEXT

In the formation of a word all characters joined together until a non-joiner occur .After the non-joiner character, a new ligature starts. This process of word formation repeated until the completion of a word. Urdu characters change their shapes based upon neighboring context, depending on whether the character joins a ligature in the initial, medial or final position, or is unconnected. Figure 1-3 shows the spelling, ligatures and the cursive form of an Urdu word respectively.

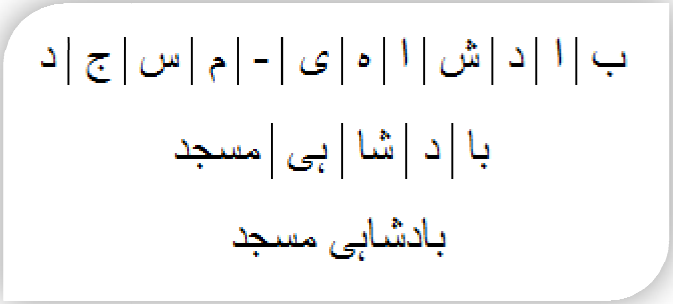

FIGURE 1-3 : SPELLING, LIGATURES AND CURSIVE WORD FORM OF A SAMPLE TEXT

1.2. INCONSISTENT USE OF SPACE

Urdu writing script does not have the concept of space to separate words. Native speakers of the Urdu language parse the sequence of ligatures into words as they read along the text. In typing, space is used to get the right character shapes and sometimes it is used within a word to break the word into constituent ligatures as shown in the Figure 1-3.In Urdu script, space do not separate the

two words rather, readers are able to distinguish the boundaries of two words from the sequence of ligatures for example "اردوخط" is distinguishable for the Urdu reader as two words.

1.3. WHAT IS A WORD?

Whenever this question comes into our mind, we take it very obvious as if we are very clear about definition of a word. But in fact, sometimes even native speakers of a language may have conflict on some words in that language. The reason behind this is the fact that there is no standard definition of a word. Usually a word is defined as a unit of language that has some meaning. It is composed of one or more morphemes which are linked more or less tightly together, and has a value phonetically. Words can be combined to create phrases, clauses and sentences [1]

In linguistics, generally a “word” is a single unit of expression and it is considered as the most stable unit which is uninterruptible by space [18].

1.4. WHAT IS WORD SEGMENTATION PROBLEM?

Some languages such as English provide the clear indication for words. In such languages the words are separated using the space. However, word segmentation problem is present in many languages like Chinese, Thai, Urdu, Arabic etc. because these languages do not have explicit boundary or delimiter such as space or comma between the words. For natural language processing word segmentation or word tokenization is preliminary task for understanding meanings of the sentences[18][19][20][21][23]. It has application in many areas like spell checking, POS, speech synthesis, information retrieval and text categorization [19] but here we study word segmentation from the point of view of Optical Character Recognition (OCR) System.

1.5. WORD SEGMENTATION PROBLEM IN OCR SYSTEM

The purpose of an OCR system is to convert a document image into an editable document. An OCR system involves a number of different processes such as pre-processing, feature extraction, training, recognition and post-processing. In each phase further different activities are performed. For example Pre-processing involves noise removal, layout analysis, skew detection and correction, identification of different runs, line detection, thinning and skeltonization etc [2] [3]. In the recognition process characters or ligatures are recognized using classifier such as neural networks, HMMs or tree classifiers. But before recognition, training is performed on the corpus and is fed into the recognition system [15].

The output of the recognizer is in the form of characters/ligatures. The next process is to define the word boundaries using these recognized characters/ligatures. This process is called word segmentation. In word segmentation recognized ligatures or characters are joined together in such a way that explicit boundaries of words are identified. Spaces are introduced in appropriate positions. Word segmentation model for the Urdu OCR system can take input in either character's form or ligatures form to make words from them. In this work, it is considered that word segmentation model obtain input in form of ligatures from the OCR recognizer. For example

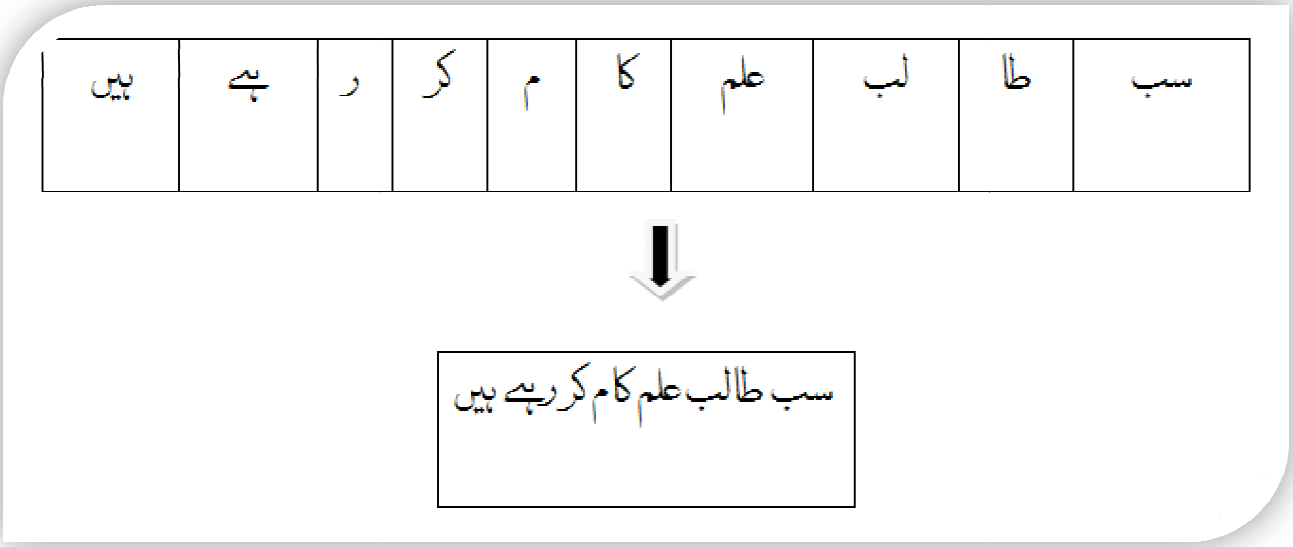

FIGURE 1-4 : EXAMPLE OF LIGATURES TO WORD FORMATION IN URDU

Other sub processes of post processing are diacritic placement and layout management. An overview of an OCR system with respect to word segmentation is given below

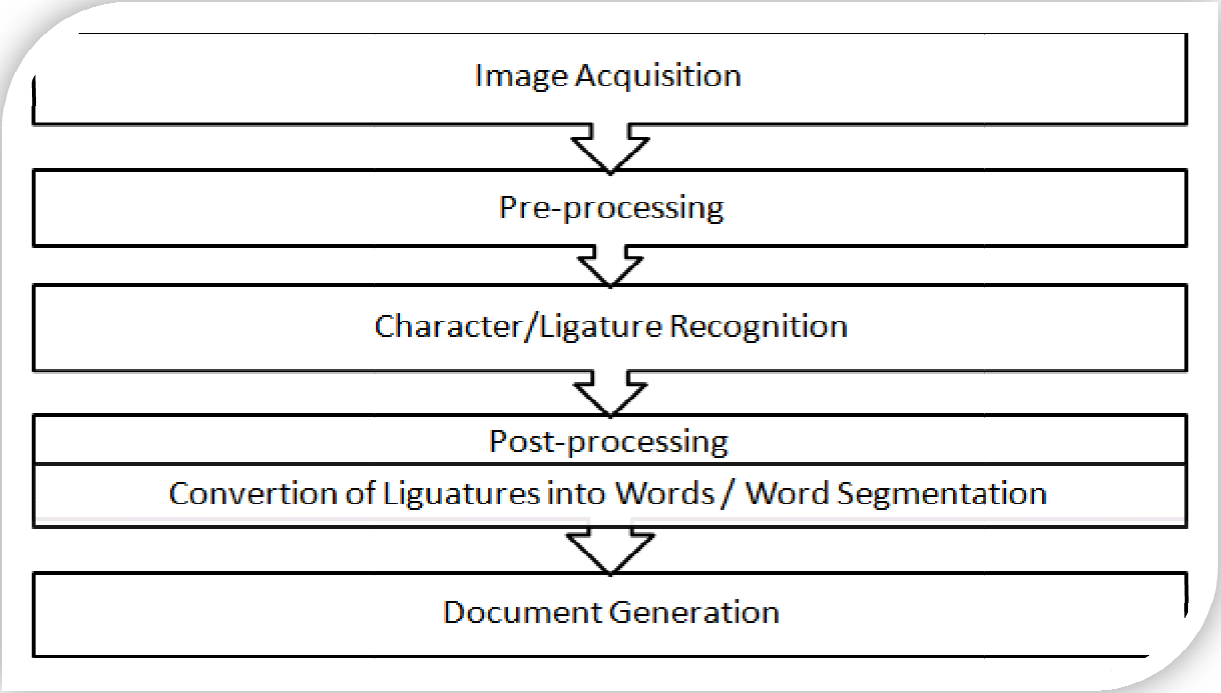

FIGURE 1-5 : OCR SYSTEM

1.6. PROBLEM STATEMENT

The purpose of this study is to solve the word segmentation problem for the Urdu OCR system. That is to convert a given sequence of ligatures into a sequence of words and resolve ambiguity among them. The solution to this problem statement will improve the overall performance of Urdu OCR System.

2. LITERATURE REVIEW FOR EXISTING TECHNIQUES

The techniques used previously for the solution of word segmentation problem in different languages are classified into the following three categories:

- Dictionary/ Lexicon based approaches
- Linguistic Knowledge Based Approach
- Machine Learning based Approaches /Statistical Approaches

The following section briefly reviews the different techniques of these categories.

2.1. DICTIONARY / LEXICON BASED APPROACHES

Dictionary based approaches (DCB) or Lexicon based approaches are efficient and straight forward [23].These approaches segment the input text into words using the dictionary or lexicon. DCB's accuracy and performance highly depend on the quality and size of the dictionary. While using techniques of this category, unknown word problem that is also known as out of vocabulary (OOV) or ambiguity problem, may occur [23]. Where unknown words are words in given text which are not available in the dictionary and ambiguity problem is due to more than one ways of segmentation for a given sequence of characters [21].Most commonly used techniques are

- Longest Matching Approach
- Maximum Matching Approach

2.1.1. LONGEST MATCHING APPROACH (LM)

Longest matching [4] is one of the earliest approaches of this category. Longest Matching scans the text from left to right (right to left for Arabic script) and finds the longest match from the dictionary by comparing text at each point. If, after the selection of word boundary, the remaining sentence does not have match to the entries of dictionary then selection process is back tracked.

The segmentation in this method can be started in any direction but [22] uses LM in forward direction with the word binding force for Chinese Word Segmentation. Since most of Chinese words are of length one or two, so a lot of time is wasted for searching its longest match. So in this technique the lexicon is divided according to length of the words and five corpus tables of length 1, 2, 3, 4, and more than 4 characters are built. For this purpose whole corpus is scanned and all the single and two characters words are stored separately in one or two character tables and if a three character word appears then it is stored in the form of two character prefix and one character suffix and also stored in the two character and one character tables respectively with the status of prefix or suffix .Similar process is performed for the 4 character word. So each entry in the corpus act as pointer to the one or two word tables with their status of affixes and infixes. Then these corpus entries are combines to find the longest match [22].

Longest Match has greedy characteristics and therefore fails in certain scenarios. For example in Thai word segmentation, Longest Match can be unsuccessful for the segmentation of ไปหามเหลี (Go to see queen). Longest Match gives segmentation as ไป (go), หาม (carry), เห (deviate), ลี (color). However the required segmentation is ไป (go), หา (see), มเหลี (queen) [4].

2.1.2. MAXIMUM MATCHING APPROACH (MM)

In Maximum matching algorithm the character strings are matched with the lexicon entries and the best segmentation among all the possible alternatives sequences is selected with the fewest and longest words. The algorithm works from left to right (right to left for Arabic script) and searches the longest matching word .If the sentence is comprised of single character words then this algorithm will give a unique solution. As the algorithm determines the segments locally so the resulting sentence segmentation is always suboptimum. Experiments of using this method reveal that the size of a lexicon is even less important than the suitability of the lexicon to the particular corpus [5].

Forward and backward MM methods are invariant of MM on the basis of the starting direction of the segmentation and work as an alternative for finding segmentation ambiguities. In the first step of MM, segmentation results are obtained by applying both forward and backward MM and in the second step common segments are selected from the two chains of words, and then apply some heuristic rules or language knowledge to resolve the conflicted segments in order to find the optimal results [23].

MM gives better results than the longest matching approach but problem arouses when alternative sentences have the same number of segments. So for this situation, best candidate is selected using some other technique or longest matching at each point technique [23].

2.2. LINGUISTIC KNOWLEDGE BASED APPROACHES

Linguistic knowledge based approaches like Dictionary based approaches also rely very much on the lexicon. Techniques in this category usually come across with all possible segmentations of a sentence in the start and then select the most likely segmentation from the set of possible segmentations using a probabilistic or cost-based scoring mechanism. For example, a simplest

approach scores all the alternative segmentations based on the word frequency and picks the sentence with the highest cost [23].These approaches diverge by their scoring or path searching processes. Some of these techniques are discussed below

- Using N-grams
- Maximum Collocation Approach

2.2.1. USING N-GRAMS

In the literature unigram, bigram and trigram were also used for the word segmentation especially for Chinese language. In [10] a lexicon is represented as a Weighted Finite State Transducer (WFST). Each word unigram value is assigned as a weight to this word in WFST and lowest cost path is selected as a best sequence of segments after the summation of the unigram cost over all the alternative possible paths. For decoding process Viterbi algorithm is used. Since lexicon does not have number of words like dates, numbers, proper name and places. In order to cater these words, a productive morphological process is built within a WFST by introducing transition weights between the bodies and their affixes, such as nouns and their plural form as a suffixes .In [11] a WFST is also proposed to detect Chinese proper names in statistical manner.

If the unigrams are used only as word segmentation tool, then segmentation ambiguity problem cannot be resolved as segmentation ambiguity cannot be resolved locally. So there is a need for contextual constraints for the appropriate segmentation to make judgment on the broader context. So the bigram and trigram are more sensible to serve the high order language models. In [23] two cases of unexpected segmentation are discussed. In the first case overlapping ambiguity might exist where a character could go either way to form two words and in the second case composition ambiguity might exist where the sub-segmentation is possible. But by using bigram and trigram these ambiguities were resolved.

In [12] an idea of constructing a word lattice from a character string given a lexicon is presented where all the possible word segmentation results are preserved. Each word is associated with a unigram. Similarly, each word transition is associated with a word or word class bigram. Viterbi algorithm is implemented to decode the best path with least cost, which take into account both word unigram and bigram and this word lattice is passed to stack decoder to have N-best list by using these grams. Due to searching space and decoding time the trigram is not used in the stack decoder at the first stage of this algorithm. This word lattice or word network is constructed in a synchronized way with a pre- assumption that any character could serve as word boundary.

There are word segmentation techniques that are derived from Viterbi framework. For example, maximum matching is an extreme case of Viterbi that keeps only one extension path when traversing forward or backward. Also Exhaustive matching includes several variations of Viterbi procedures under various searching criteria, for example

- Minimum segmentation is a Viterbi procedure under least word transition criterion
- Maximum word length is under maximum average length rule [13] [14]

2.2.2. MAXIMUM COLLOCATION APPROACH

In literature maximum collocation approach is presented for word segmentation of Thai language. The researches reveal that the main problem of improper word extraction is basically improper syllable extraction. In the technique presented in [16], an idea of performing segmentation as syllable segmentation rather than word segmentation is used. As syllable is better defined unit and a consistent syllable corpus is easy to build. So proposed word segmentation is composed of two phases: In the first phase syllables are extracted using trigram statistics and in the second phase these syllables are merged using collocation between them.

Thai grammars describe words as combination of syllables. These syllables give different meanings in isolation but when they are joined with other syllables they give different meanings. In Thai,

words are distinguished as simple words and compound words. Simple word can have one or more syllables and the meaning of each syllable can be entirely different from the whole word. The compound word is the combination of two or more words. Each word may have entirely a different meaning from the composed words.

A Thai syllable is composed of vowel form, initial consonant and final consonant. All Thai syllable patterns can be determined and list down by a little effort. The number of these patterns is finite. The direct application of identified patterns on the strings can lead to ambiguities but if the trigram statistics of syllable is applied, then words can be segmented correctly. A training corpus is composed of 553,372 manually segmented syllables that are gathered from newspapers. Viterbi algorithm is used in [16] for the best segmentation results and up to 99.8 % accuracy is achieved.

In syllable merging process the boundaries which can be removed from the syllable segmented sentences were determined and remaining boundaries are considered as word boundaries. The first approach is based on collocation strength between the syllables to merge syllables. Collocation here means co- occurrences of syllables observed from the training corpus and it is assumed that if a word has two or more syllables then these syllables will always co–occur. So these syllables have higher collocation than the syllables that are not part of the word. But for a corpus this collocation strength is always constant and some other approaches are also required to assist it. So lexical knowledge obtained from dictionaries is used to decide the given sequence of syllables is a word, dictionary look up is used. Then the overall collocation strength of the sentence is measured. This can act as force to put the syllables together. There can be a driving force which stops the syllables to occur together. So over all collocation strength is sum of the collocation within the word minus the collocation strength between the words. Maximum collocation strength obtained is resulted in best segmentation. This method also called max Coll A method.

This paper presents two different variations in the Coll A model. In first variation only those syllables collocation is subtracted which is further part of another word. This variation is called Max Call-B. Second variation named Max Call-C does not perform any subtraction of syllables.

The corpus used for testing of MaxColl-A, MaxColl-B, MaxColl-C and MaxMatch , consists of 20,498 syllables .These algorithms give 96.3 % ,97.97 % ,98.02 % , 98.56 % precision respectively. Over all MaxColl-C performed better than the other algorithms [16].

2.3.MACHINE LEARNING BASED APPROACHES /STATISTICAL APPROACHES

Machine learning based techniques apply learning algorithms that define a function from a domain of input samples to a range of output values. These approaches mainly use a corpus in which word boundaries are explicitly marked. These machine learning algorithms build statistical models based on the features of words surrounded by the boundaries. These approaches do not require dictionaries and unknown word and ambiguity problems are handled by extracting sufficiently rich contextual information and by providing a sufficiently large set of training examples to enable accurate classification [6]. Overview of the machine learning approaches is given below

- Word Segmentation Using Decision Trees Approach
- Word Segmentation Using Lexical Semantic Approach

2.3.1. WORD SEGMENTATION USING DECISION TREES APPROACH

Thanaruk in [18] gives the idea of the word segmentation for Thai language on basis of Thai Character Cluster (TCC). Thai Character Cluster (TCC) is indivisible unit of the connected characters and segmentation of text into TCC is much easier than word segmentation. This method of segmentation is proposed in [7]. In [18] word segmentation process is performed in two sub-stages.

In first stage the text is segmented into TCCs and in the second stage Decision tree is used to combine the TCCs into words.

Segmentation of text into TCCs is performed by applying the set of rules (for example 42 BNF rules). This method does not require a dictionary and it correctly segments the text at each word boundary. The accuracy of this process is 100% in a sense that the resultant TCCs cannot be further divided and these TCCs are sub strings in two or more words.

For learning process of decision tree some attributes are defined for identifying whether two adjacent TCCs are combined to one unit. This paper presents eight attributes on which decision can be made. These are front vowel, front consonant, middle vowel, middle consonant, rear vowel, rear consonant, length, space and enter. The obtained training set is used as input to C4.5 application [8] for learning of decision trees. At each node of tree the final decision making factor is calculated by number of terminal classes. For experiment TCC corpus is divided into training and testing corpus. Results show that the method proposed in this paper gives the reasonable percentage of accuracy, precision and recall. For experiments, the best level of permission for highest accuracy is approximately equals to 70%, which gives the accuracy equal to 87.41%.

In [17] automated word extraction technique is proposed for word extraction which will list acceptable Thai words using decision trees. The approach used C4.5 [8] decision tree induction program for learning algorithm of word extraction. Thai language processing is based on information acquired from human made dictionaries and has drawbacks like these dictionaries do not handle a word not registered in dictionary and also fail to cover all words appear in corpus. This algorithm iteratively analyzes the contents of the list of attributes and builds a tree from these attribute values where leaves of the tree represent desired goal attributes. In each step branch of the tree is decided using highest information obtained, all the training data set is classified. C4.5 algorithm recursively analyzes and determines whether expected error rate can be minimized by replacing a leaf or a branch with another leaf or branch.

Word extraction problem is solved by distinguishing a word string from the non-word string on the basis of following attributes. These attributes are used for learning algorithm. The first attribute used for word extraction is left and right mutual information where the mutual information is the ratio of probability of co-occurrence of a and b to the independent probability of co-occurrence of a and b. High mutual information means a and b can co–occur more than expected value. If xyz is a word then both Lm (Left Mutual Information) and Rm (Right Mutual Information) of xyz should be high otherwise xyz is a non-word and consists of words and characters.

Other two attributes of word extraction are left and right entropy. Entropy is a measure of disorder of a variable. If y is a word then alphabets proceeding it and following it should have varieties or high entropy but if it is not a complete word then its left or right words has less varieties and its entropy must be low.

Next attributes used for C4.5 algorithm, for word extractions are frequency of words and length of strings. The frequency of words should be higher than those of the non-words strings. For obtaining independent frequency of words its occurrence is divided by the size of corpus and its value is multiplied by the average value of the Thai word's length. Functional words for example ‘will’ or ‘then’ can mislead the occurrences of the word’s patterns so these words are filtered out from text.

The next attribute verifies that given word is of correct spelling or not. For application of C4.5 algorithm for Thai word extraction process firstly a training set is constructed. Then attributes of the strings are computed and then these strings are tagged as words or non-words. These tagged words and their attributes are used as sample for learning algorithm. From this training data a decision tree is constructed. The precision of the algorithm is 87.3 % for training set and 84.1 % for test sets. The recall of the extraction process is 56% for both training and test sets. The results indicate that this accuracy can be further enhanced if a larger corpus is used with longer strings. The results obtained from this experiment are compared with the results gained form Thai Royal

Institute Dictionary (RID). The created decision tree performed better than RID and it turned out to be vigorous for unseen data as well. 30% extracted words are not found in RID.

2.3.2. WORD SEGMENTATION USING LEXICAL SEMANTIC APPROACH

All these above motioned methods do not consider the semantics of Thai language for word segmentation .Method proposed in [20] consider semantics of the language as well and execute word segmentation approach in four stages. These stages are: generating all the possible candidates, proper noun consideration, semantic tagging and semantic checking. This technique used the word hierarchy which classifies words by their meanings. Each word is associated with a group of meaning called "A Kind Of" (AKO) and it is used to analyze the meanings of sentence and to reduce ambiguities in sentences. 74 sub categories of the AKO number are identified in this paper for example category one is "concrete" which is further sub divided into subject as person or organization and concrete place as region and natural place.

For this purpose a semantic corpus is constructed using the semantic information to distinguish each word. The meaning of each word is in AKO number form. For this purpose ORCHID [8] syntactic semantic corpus is used and AKO number are added. Then in the first stage of word segmentation approach, forward and backward maximal matching algorithms are used for generating all possible words using dictionary. In the second stage the word segments obtained from the first stage are compared with the human tagged words. In the Semantic tagging stage each word is labeled with an AKO number for example word 'birthday' is tagged with 'Time' and 'celebrate' is tagged with 'Action'. If the semantic patterns of sentences are same then the selection is performed on the priority of proper noun. In the semantic checking stage using semantic corpus the frequency of patterns is computed and assigned as semantic score to it and the results with highest priority of proper noun and highest score are selected. This technique gives the 97.3% accuracy of the word segmentation.

2.4. FEATURE BASED APPROACH

A feature can be anything that tests for specific information in the context around the target word sequence. In the feature based approaches word segmentation problem is treated as word sequence disambiguation problem [24]. In the feature based approaches several type of features are employed but for this word segmentation task context word features and collocation features are considered more important. Context based features are used to test occurrence of a particular words within +/- k words of the target word sequence and collocation features are used to test the text patterns for only two contiguous words and/ or the part of speech tags around the target word [25]. For automatically extraction of these features two learning algorithms are purposed. These are:

- Winnow
- RIPPER

2.4.1. WINNOW

In Winnow algorithm a network named as “winnow” is constructed which is composed of several nodes connected to a target node. Each node called as “specialist”, of this network owns a particular value of an attribute and on the basis of its specialty, it also votes for a value of target concept. Then this algorithm combines the vote form all specialists and makes a prediction based on weighted-majority votes [25]. If this algorithm fails in prediction then the weight of the specialist that predicts incorrectly will be moved down and the weight of the specialist that predicts correctly will be promoted [26].

2.4.2. RIPPER

RIPPER learning algorithm is a propositional rule learning algorithm that builds a rule set which classifies the training data. It has rules of form like

If (T1 and T2 and … Tn)

Then class Cx .

Where Tis are set of conditions that are tested for particular value of an attribute and Cx is the target class to be learned. Following table shows the comparison results of the both techniques and taken from [25].

	Context Independent		Context Dependent	
	Training Set (%)	Test Set (%)	Training Set (%)	Test Set (%)
Maximal Matching	79.74	78.85	52.10	53.52
Trigram	99.81	99.77	73.30	73.15
FEATURE-1-RIPPER	99.94	99.74	96.98	86.60
FEATURE-1-Winnow	100.00	99.70	100.00	95.33
FEATURE-2-RIPPER	98.52	91.27	93.28	89.00
FEATURE-2-Winnow	99.97	93.82	100.00	92.10

TABLE 2-1: THE RESULT OF COMPARING DIFFERENT APPROACHES [25]

For the both of these algorithms a corpus of 25,000 sentence is used which also includes ambiguous strings. In this corpus each paragraph is separated into sentences and then into words and each word is manually assigned an appropriate POS tag by linguists. The performance of both algorithms is measured by the percentage of the number of correctly segmented sentences to the total number of sentences. As given in the performance table 1, both RIPPER and Winnow have capability to construct rule sets or networks that extract the features from data effectively and are able to capture useful information that cannot be found by traditional word segmentation model such as trigram, and make the task of word segmentation more accurate.

3. METHODOLOGY

The methodology followed for the solution of Urdu word-segmentation problem is similar to build a language model that is, to use the ligature co-occurrence information along with words collocation information to construct a language model. In order to execute this methodology, we have built a proper segmented training corpus.

The whole process is completed in three phases. In the first phases, data necessary for the Urdu Word Segmentation model is collected and using this collected data ligature and word probabilities are calculated. For this purpose firstly some cleaning issues are resolved and then these probabilities are calculated. Figure 3-1 shows the execution flow of this phase.

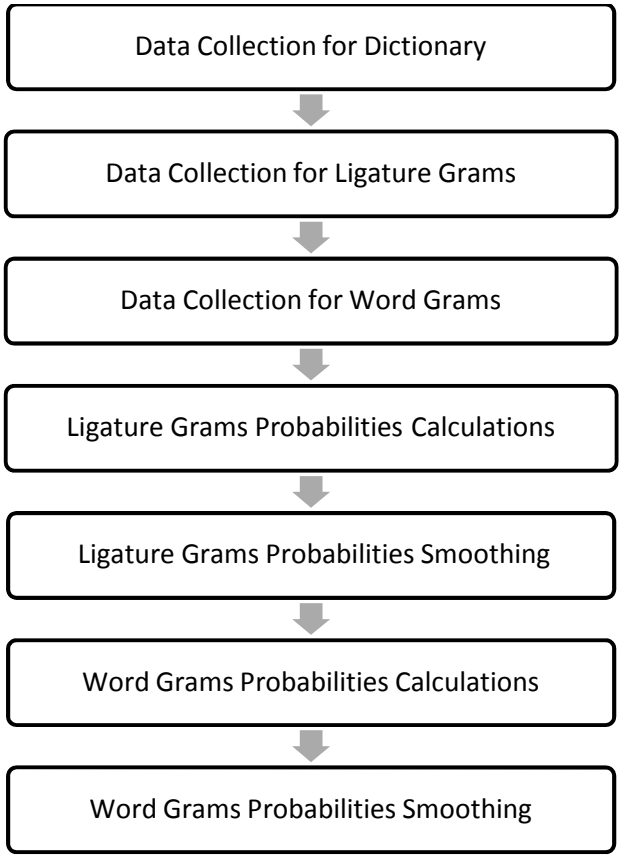

FIGURE 3-1 : EXECUTION FLOW OF THE FIRST PHASE (DATA COLLECTION AND PROBABILITIES CALCULATIONS)

In the second phase, from input set of ligatures, all sequences of words are generated and ranking of these sequences is performed using the lexicon lookup. According to a selected beam value, top *k*

sequences, with more valid words heuristic, are selected for further processing. Figure 3-2 represents the completion flow of the second phase.

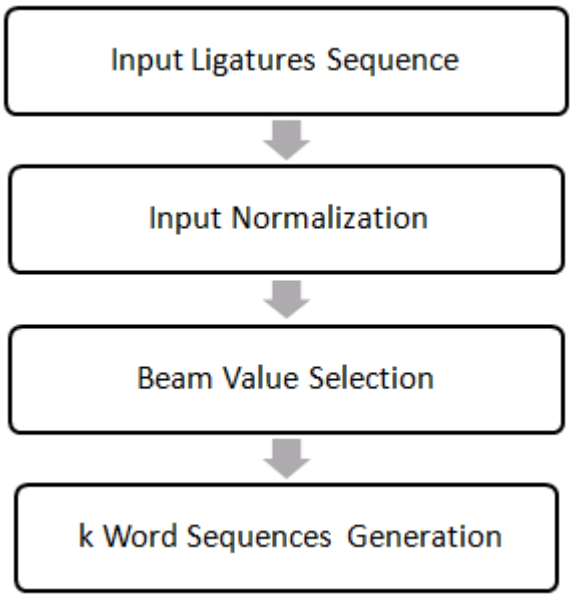

FIGURE 3-2 : EXECUTION FLOW OF SECOND PHASE (GENERATION OF K WORD SEQUENCES)

In the third phase, maximum probable sequence, from these *k* word sequences is obtained using all variation of the technique presented in section 6. The word sequence which is suggested by most of these techniques, as maximum probable sequence, is selected as an optimal word sequence for the input ligature sequence. The execution flow for the third phase of methodology is given below in the figure 3-3.

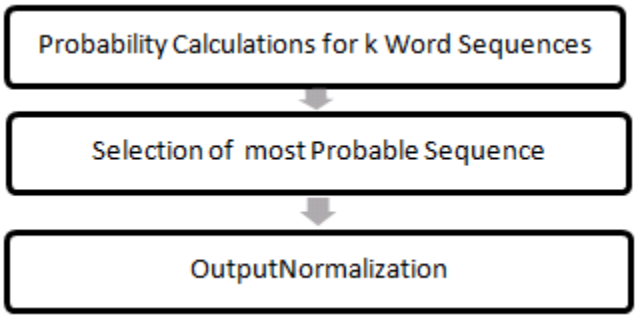

FIGURE 3-3 : EXECUTION FLOW FOR THE THIRD PHASE (SELECTION OF OPTIMAL WORD SEQUENCE)

Details of above three phases are described in subsequent sections.

4. DATA COLLECTION AND PROBABILITIES CALCULATIONS

This step involves collection of data to be used for the word segmentation model. Most of the data is collected from the Center for Research in Urdu Language Processing (CRULP). The whole data is used for different processes in the word segmentation model. This data involves

- Data for building a word dictionary
- Data for the ligature grams
- Data for the word grams

The detail of each of the above data is given below.

4.1. DATA FOR BUILDING A WORD DICTIONARY

For building a dictionary we have collected the Urdu words from all domains which cover affixes, person names, countries and cities names and company names. We have obtained these lists from CRULP. A clean-up process is required for above data to be used for our purpose. The details of data and their clean-up process are as follows.

- A distinct word list of 50169 words is obtained. This word list is generated from the 18 million word corpus and after manual cleaning of word list we have obtained word list of 49630 unique words after removing words which do not exist as a valid word in Urdu online dictionary [28]. For example words like ابڑا،ادبا etc are removed from the word list.

- The affixes list which is added to the word dictionary is also modified by insertion of the zero-width-non-joiner. This list is also maintained without zero-width-non-joiner for further processing in data word grams. Following table 4-1 shows some examples of affix words which require a zero width non joiner (ZWNJ).

Affix words with Space	Affix words with ZWNJ
اجتماع گاه	اجتماع‌گاه
احاطه گیری	احاطه‌گیری
احساس کده	احساس‌کده
احسان مند	احسان‌مند
طلاق شده	طلاق‌شده
طلاق یافته	طلاق‌یافته
طوفان زده	طوفان‌زده
عقیدت مندانه	عقیدت‌مندانه
عمارت سازی	عمارت‌سازی
غم کده	غم‌کده
غنڈه گردی	غنڈه‌گردی

TABLE 4-1: EXAMPLE OF AFFIX WORD WITH SPACE AND ZWNJ

- Similarly, countries and cities names with spaces are joined with zero-width-non-joiner (ZWNJ) character and added to the dictionary and also maintained with space for further processing in data grams. Table 4-2 shows some examples of countries and cities names which require a zero width non joiner (ZWNJ).

Affix words with Space	Affix words with ZWNJ
آرم سٹرانگ	آرم‌سٹرانگ
آستان اشرفیه	آستان‌اشرفیه
آسترسند	آسترسند
اسلام آباد	اسلام‌آباد
اشک آباد	اشک‌آباد
اوون ساؤنڈ	اوون‌ساؤنڈ

Affix words with Space	Affix words with ZWNJ
بام پور	بام‌پور
چاہ بہار	چاه‌بهار
خرم آباد	خرم‌آباد
حسن ابدال	حسن‌ابدال
حیدرآباد	حیدرآباد
آئرلینڈ	آئرلینڈ
بنگلہ دیش	بنگله‌دیش
گوئٹے مالا	گوئٹے‌مالا

TABLE 4-2: EXAMPLES OF CITIES AND COUNTRIES NAMES WITH SPACE AND ZWNJ

- Person names and company names are tokenized on space and added as words in the dictionary.

Table 4-3 shows counts of all above categories in the dictionary.

Distinct word list	Affixes list	Person Names	Brand Names	Countries Names	Cities Names	Tourist Places	Total words
49630	2027	20432	734	279	1938	187	70420

TABLE 4-3: TABLE SHOWING THE COUNTS OF CATEGORIES IN OUR DICTIONARY

4.2. DATA COLLECTION FOR THE LIGATURE GRAMS

The Corpora used for building ligature grams consists of half million words. This corpus is collected from the Center for Research in Urdu Language Processing (CRULP).CRULP has a raw corpus of 18 million words of Urdu text alienated domain-wise, mostly collected from Jang News and BBC Urdu service [30]. For this project, from 18- million word corpora, 300,000 words are taken from Sports,

Consumer Information and Culture/Entertainment domains. 100,000 words are obtained from Urdu corpus available at [31] from the project of Urdu-Nepali-English Parallel Corpus. 100,000 words are obtained from Hassan's POS tagged Corpus [32] .Tags of this corpus are removed before further processing.

4.3. DATA COLLECTION FOR THE WORD GRAMS

For the computation of word grams, a corpus is obtained which is comprised of the 18 million words of Urdu text. The details of the word corpora related to different domains is as follows

Domains	Cleaned Corpus	
	Total Words	Distinct Words
C1. Sports/Games	1529066	15354
C2. News	8425990	36009
C3. Finance	1123787	13349
C4. Culture/Entertainment	3667688	34221
C5. Consumer Information	1929732	24722
C6. Personal communica-tions	1632353	23409
Total	18308616	50365

TABLE 4-4 : DISTRIBUTION OF URDU CORPUS DOMAIN WISE FOR WORD GRAMS [30]

4.4. LIGATURE GRAMS PROBABILITY CALCULATIONS

For calculating the ligature grams, a cleaned properly segmented ligature corpus is required. Therefore before converting the word corpus to ligature corpus, a half million words corpus is cleaned for proper segmentation. As Cleaning a corpus is very monotonous and time consuming task and cleaning merely with manual effort is very slow. Therefore, the corpus cleaning for ligature grams included some automated tasks but most of the work is done manually.

4.4.1. CLEANING OF LIGATURE CORPUS

Since basic source for Sports, Consumer Information and Culture/Entertainment corpora files is newspaper so these files are cleaned to remove hypertext markups and English characters. As described before the "space character" in Urdu script has been used between the two words to correct the glyph shaping, not to separate the words. Therefore collected Urdu corpora have problem of space insertion, space removal and insertion of Zero-width-non-joiner (ZWNJ) to maintain the correct shape of words. Examples of these words from Urdu Corpora are given in following table. In first column of Space Removal "-"indicates space character *.

Without and with ZWNJ Insertion		Without and with Space Insertion		Without and with Space Removal	
اللہ تعالی	اللہ‌تعالی	ازکملاکھ	ازکم لاکھ	انگلینڈ	ا-نگلینڈ
پناہگزین	پناہ‌گزین	اسکا	اس کا	اتھارٹی	اتھا-رٹی
پوسٹمارٹم	پوسٹ‌مارٹم	اسکو	اس کو	اثرانداز	اثر-انداز
ترقیپذیر	ترقی‌پذیر	اسلئے	اس لئے	افسروں	افسر-وں
ذخیرہاندوزوں	ذخیرہ‌اندوزوں	اسمیں	اس میں	ائرفورس	ائر-فورس
ذخیرہشدہ	ذخیرہ‌شدہ	اعتمادرنز	اعتماد رنز	آٹوموبائل	آٹو-موبائل
ذمہداری	ذمہ‌داری	اعظموزیر	اعظم وزیر	آثارقدیمہ	آثار-قدیمہ
علامہ اقبال	علامہ‌اقبال	اقبالاسٹاف	اقبال اسٹاف	خودغرضی	خود-غرضی
عہدہبرآ	عہدہ‌برآ	اورآنے	اورآ نے	خودکار	خود-کار
فائدہمند	فائدہ‌مند	پاکفوج	پاک فوج	خودکش	خود-کش
قانونسازی	قانون‌سازی	پڑاہے	پڑا ہے	زیرانتظام	زیر-انتظام
اسلامآباد	اسلام‌آباد	کرچکا	کر چکا	صاحبزادی	صا-حبزادی
منصوبہبندی	منصوبہ‌بندی	کردیا	کر دیا	ضروریات	ضروریا-ت

TABLE 4-5: A TABLE SHOWING EXAMPLES OF ZWNJ INSERTION, SPACE INSERTION AND SPACE REMOVAL FROM THE CORPORA

We have obtained an initial space insertion list of from CRULP recourse and used this list in the process of corpus cleaning. This process works recursively for each 100 thousand words text file in the corpora as follows

1. Space insertion, space removal and ZWNJ insertion lists are applied on given text file, if these lists are available.
2. The word bigrams for this text file is generated.
3. ZWNJ insertion list, space removal list and space insertion list are created from word bigrams by the manual analysis.
4. Generated text file is modified as
    4.1. Space is inserted in a word if that word exists in space insertion list.
    4.2. Space is removed between two words to make it a single word by the use of space removal list.
    4.3. ZWNJ character is inserted in a word, to correct the shape of character glyph in that word by the use of ZWNJ insertion list generated in step 3.
5. Using a file comparer, updated file created in step 4 is compared with original text and changes are highlighted. Then only highlighted strings are considered and corrected manually, if needed, according to the context of these strings.
6. Current space insertion list is merged with the previously available space insertion list.
7. Current ZWNJ insertion list and space removal list are merged with existing lists as well, if these lists are available in previous iteration.
8. Next iteration is started again from step 1 for the next corpus file.

These iterations resulted in cleaned corpus files with the same names as original corpus files.

4.4.2. CONVERSION OF WORD CORPUS TO LIGATURE CORPUS

Ligature is a sequence of characters in a word separated by non-joiner characters or the Unicode ZWNJ character. Figure 1-2 gives the list of Non-joiners. These Non-joiners appear at only isolation and final position. The algorithm of converting the word corpora to the ligature corpora is as follows

```
For each character in the Input text file
        If this character is a non-joiner
                Append this character to the output text file with space
        Else
                Append this character to the output text file
        End If
End For
```

FIGURE 4-1: PSEUDO-CODE FOR WORD TO LIGATURE CONVERTION

Using the above pseudo code the word corpora collected for ligature grams is converted to ligature corpora. A ligature unigram is a distinct ligature in a corpus.

4.4.3. LIGATURE UNIGRAMS, BIGRAMS AND TRIGARMS PROBABILITY CALCULATIONS

For calculating the ligature grams from ligature corpus, space is also considered as a separate ligature which let us know the exact boundaries of the ligatures from where the words end and from where the word starts.

To distinguish the boundaries more accurately we build the ligature corpus with the double space and construct language trigram model using this double spaced corpora. The reason behind this is to know the probabilities of the ligature with which words start and end.

A ligature unigram is a distinct ligature in a corpus and its frequency is equal to the number of occurrences of that ligature in the corpus. For example for the sentence

"تم ۔ یہاں ۔ کیوں ۔ کھڑے ۔ ہو"

The Ligature Frequencies are as follows

۔	ہو	ے	کھڑ	کیو	ں	یہا	تم
567387	9571	5449	100	667	15324	283	52

TABLE 4-6: LIGATURE FREQUENCIESOF SMAPLE TEXT

Unigram probability of ligature is equal to the frequency of that ligature in the corpus divided by the total number of ligatures of the corpus. Ligature unigram probability this can be represented mathematically as

$$P(l_i) = \frac{C(l_i)}{\text{Total Number of Ligatures}(N)} \quad (1)$$

For the above sample text the unigram probabilities of ligatures using equation (1) are given in the Table 4-7 as follows

۔	ہو	ے	کھڑ	کیو	ں	یہا	تم
0.376231866	0.006346489	0.003613208	6.63E-05	0.000442285	0.010161278	0.000187656	3.45E-05

TABLE 4-7: PROBABILITIES OF LIGATURES FOR THE SAMPLE TEXT

A ligature bigram model approximates the probability of a ligature given the previous ligatures by using the conditional probability of preceding ligature [27]. Mathematically it can be represented as

$$P(l_i|l_{i-1}) = \frac{C(l_{i-1}l_i)}{C(l_{i-1})} \quad (2)$$

Table given below shows the bigram probabilities for the above example.

	-	ہو	ے	کھڑ	کیو	ں	یہا	تم
-	0.169172011343228	.02552769070 01045	.0940528634 361233	0	.00525525525 525526	.09770912423 09203	0	1
ہو	.0 015849852 0410231	0	0	0	0	0	0	0
ے	.5 287396433 12237E-06	.0 0123301985 370951	0	0.24	0	0	0	0
کھڑ	.0 000174484 082293038	0	0	0	0	0	0	0
کیو	.0 000942919 030573489	0	0	0.02	0	0	0	0
ں	0	.0 0676071055 3814	0	0	.0 2222222222 22222	0	.0 81625441 6961131	0
یہا	.0 000271419 683566948	0	0	0	0	0	0	0

تم	.7 226108458 60057E-05	0	0	0	0	0	0	0

TABLE 4-8: BIGRAM PROBABILITIES FOR THE SAMPLE TEXT

Since some probability values are zero so it requires smoothing. The smoothing technique is discussed in Section 4.5.3.1.The probabilities of the bigram ligatures in the above sentence are as follows

Probability	Bigram Ligature
1	تم-
000271419683566948.0	-یہا
816254416961131.0	یہاں
000942919030573489.0	-کیو
222222222222222.0	کیوں
977091242309203.0	-ں
24.0	کھڑے
940528634361233.0	-ے
0158498520410231.0	-ہو

TABLE 4-9: PROBABILITIES OF THE BIGRAM LIGATURES FOR THE SAMPLE SENTENCE

Similarly a ligature trigram model approximates the probability of a ligature given the previous ligatures by using the conditional probability of preceding two ligatures [27].Mathematically it can be represented as

$$P(l_i|l_{i-1}) = \frac{C(l_{i-2}l_{i-1}l_i)}{C(l_{i-2}l_{i-1})} \quad (3)$$

For the sentence with the double space

"تم - - یہاں - - کیوں - - کھڑے - - ہو"

The trigram probabilities calculated from ligature corpora is given below

Probability	Bigram Ligature
1	تم - -
000271419683566948.0	- - یہا
1	- یہاں
978354978354978.0	یہاں -
1	ں - -
000942919030573489.0	- - کیو
207476635514019.0	- کیوں
966216216216216.0	کیوں -
1	ں - -
000174484082293038.0	- - کھڑ
242424242424242.0	- کھڑے
1	کھڑے -
1	ے - -
0158498520410231.0	- - ہو

TABLE 4-10: TRIGRAM PROBABILITIES OF SAMPLE SENTENCE WITH DOUBLE SPACE

Once all the frequencies are calculated, next phase is to calculate the unigram, bigram and trigram probabilities of the ligature corpus to be used in word segmentation model. These probabilities are calculated firstly by using equation 1, 2 and 3 but after smoothing it is calculated using equation 6. Following table shows the count of unigram, bigram and trigram frequencies and probabilities of ligature corpora.

Ligature Tokens	Ligature Unigram	Ligature Bigrams	Ligature Trigrams
1508078	10215	35202	65962

TABLE 4-11: TABLE SHOWING THE COUNT OF UNIGRAM, BIGRAM AND TRIGRAM FREQUENCIES AND PROBABIBILITY OF THE LIGATURE CORPUS

4.5. WORD GRAMS PROBABILITY CALCULATIONS

In the calculation of ligature grams as described in previous section we first clean the corpus and then computed frequencies and probabilities of ligature grams from it. But for the word grams, the corpus is very huge and it is not possible to clean the 18 million word corpus before these calculations so some heuristics are used to clean the unigram, bigram and trigram frequencies computed from 18 million word corpora.

4.5.1. WORD UNIGRAMS, BIGRAMS AND TRIGRAMS FREQUENCIES

A unigram frequency of a word is the count of occurrences of that word in a corpus. A word unigram does not look at the context of the word in a sentence. To handle this drawback of unigrams we have bigrams. A bigram frequency is calculated for two consecutive words and it is the count of occurrences of two words together. To handle the broader context we have trigrams and if we have three words XYZ then the count of occurrences of XYZ together, in the corpus give us the trigram frequency of XYZ words.

Following table gives us the count of unigram, bigrams and trigram frequencies and probabilities of the words corpora

Word Tokens	Word Unigrams	Word Bigrams	Word Trigrams
17352476	157379	1120524	8143982

TABLE 4-12: COUNT OF UNIGRAM, BIGRAM AND TRIGRAM FREQUENCIES AND PROBABILITIES

Given below are few examples of word unigram, bigram and trigram frequencies respectively for the sentence

"جب زلزله شروع ہوا اس وقت میں دفتر میں ہی تھا"

The word unigram frequencies are

Frequency	Word Unigram
45179	جب
4740	زلزله
16396	شروع
25952	ہوا
243046	اس
25757	وقت
550000	میں
1593	دفتر
550000	میں
46513	ہی
59048	تھا

TABLE 4-13 : EXAMPLE OF UNIGRAM WORDS AND THEIR FREQUENCIES IN THE WORD CORPORA FOR A SENTENCE

Table 4-14 shows the bigram words and their frequencies in the word corpora for the sample sentence.

Frequency	Word Bigram
33	جب زلزله
2	زلزله شروع
778	شروع ہوا
361	ہوا اس
11211	اس وقت
937	وقت میں
31	میں دفتر
484	دفترمیں
1848	میں ہی
226	ہی تھا

TABLE 4-14 : EXAMPLE OF BIGRAM WORDS AND THEIR FREQUENCIES IN THE WORD CORPORA FOR A SENTENCE

Table 4-15 shows the trigram words and their frequencies in the word corpora for the above sample sentence.

Frequency	Word Trigram
2	جب زلزله شروع
4	زلزله شروع ہوا
25	شروع ہوا اس
27	ہوا اس وقت
116	اس وقت میں
1	وقت میں دفتر

Frequency	Word Trigram
1	میں دفتر میں
2	دفتر میں ہی
5	میں ہی تھا

TABLE 4-15: EXAMPLE OF TRIGRAM WORDS AND THEIR FREQUENCIES IN THE CORPORA FOR A SAMPLE SENTENCE

4.5.2. CLEANING OF WORD UNIGRAM, BIGRAM AND TRIGRAM FREQUENCIES

After calculation of the word unigram bigram and trigrams counts, following cleaning issues of corpus are handled with the help of these calculations.

4.5.2.1. HANDLING SPACE INSERTION ERROR WORDS

We have certain words which are made up of two individual words and occur with very high frequency in the corpus for example "ہوگا" exist as single word rather than two separate words in the word corpus. To solve this problem following processing is performed.

- Firstly, a list of about 700 words is made from the word unigrams. These words have frequency greater than 50. The words in this list have space insertion error that is two words are combined without space and need to exist as separate words.
- Each word of the list is manually viewed and space is inserted, where required, in each space insertion error word.
- After that these error words are removed from the word unigram frequency list and added to the word unigrams as individual words with frequency of the respective error word.

- For the word bigrams, each error word in joined word list is checked. If any of these error words is contained by a bigram word for example "کیا ہوگا" exists in the bigram list and contain "ہوگا" error word. Then this bigram entry "کیا ہوگا"is removed from the bigram list and frequencies of "ہو گا" and "کیا ہو" are increased by the frequency of "کیا ہوگا".If these words do not exist in the word bigram frequency list then these are added as a new bigram word with the frequency of "کیا ہوگا".

- Same procedure is performed for the word trigrams.

4.5.2.2. HANDLING AFFIXES ERRORS

The second main issue is the word affixes. These are treated as separate words and exist as bigram entries in the list rather than a unigram entry. For example "صحت مند" exits as a bigram entry but in Urdu it is treated as a single word. To cope with this problem following solution is applied

- The list of affixes (used in making dictionary in section 4.1) is used.
- If any entry of word bigram matches with an affix word, then this word is combined by removing space from it and inserting zero-width-non-joiner, if required to maintain its glyph shape.
- Now we inserted this word in the unigram list with its original bigram frequency.
- Same procedure is performed if a trigram word matches with an affix then it is removed from trigram and added as bigram entry with it respective trigram frequency.

4.5.3. WORD UNIGRAM, BIGRAM AND TRIGRAM PROBABILITY CALCULATIONS

Unigram, Bigram and Trigram probabilities are calculated by using following formulas respectively

$$P(w_i) = \frac{C(w_i)}{total\ Number\ of\ Words(N)} \quad (4)$$

$$P(w_i|w_{i-1}) = \frac{C(w_{i-1}w_i)}{C(w_{i-1})} \quad (5)$$

$$P(w_i|w_{i-2}w_{i-1}) = \frac{C(w_{i-2}w_{i-1}w_i)}{C(w_{i-2}w_{i-1})} \quad (6)$$

But after calculation of probabilities we came to know that smoothing is required to avoid the data sparseness. So smoothing technique is presented in the next section.

4.5.3.1. SOMOTHING OF PROBILITIES

Smoothing is a technique essential in the construction of n-gram language models. A language model is a probability distribution over strings P(s) that attempts to reflect the frequency with which each string *s* occurs in natural text. While smoothing is the central issue in the language modeling, different techniques are available in the literature but here we have chooses method One Count describe in [33] for smoothing of our language model. Using this technique estimated probabilities are calculated with the following equation

$$Pone\left(w_i \left| w_{i-n+1}^{i-1}\right.\right) = \frac{C\left(w_{i-n+1}^{i}\right)+\alpha\ Pone\left(w_i \left| w_{i-n+2}^{i-1}\right.\right)}{C\left(w_{i-n+1}^{i-1}\right)+\alpha} \quad (7)$$

$Where\ \alpha = \gamma\left[n_1\left(w_{i-n+1}^{i-1}\right) + \beta\right]$ and $n_1\left(w_{i-n+1}^{i-1}\right) = \left|w_i : C\left(w_{i-n+1}^{i}\right) = 1\right|$ and β and γ are constants.

This Pone Smoothing technique merges two perceptions. First one is that Pone (7) is a reasonable form of smoothed distribution as argued by MacKay and Peto [34] that is, the parameter α represents the number of counts being added to the given distribution and the new counts are distributed to the lower order distributions by recursive part of the equation (7).

Second institution is from the Good-Turing estimate [35]. Good-Turing describes that it can be inferred that the number of these extra counts that is denoted by α should be proportional to the number of words with exactly one count in the given distribution. This inference of the Good-Turing works well in the equation (7) as described above.

And if the equation (7) is simplified for n=3,2,1,0 for trigrams, bigrams and unigrams the resultant trigram , bigram and unigram probability estimate equations are given below respectively

Trigram Probability Estamate $=$

$$Pone(w_i \,|w_{i-2}\ w_{i-1}\ ) \quad = \frac{C\left(w_{i-2}^{i}\right)+\alpha\ Pone(w_i\,|w_{i-1})}{C(w_{i-2}^{i-1})+\alpha} \tag{8}$$

$Where\ \alpha = \gamma[\mathrm{n}_1(\mathrm{w}_{\mathrm{i}-2}, \mathrm{w}_{\mathrm{i}-1}) + \beta]$ and $\mathrm{n}_1(\mathrm{w}_{\mathrm{i}-2}, \mathrm{w}_{\mathrm{i}-1}) = |\mathrm{w}_{\mathrm{i}}: \mathrm{C}(\mathrm{w}_{\mathrm{i}-2}, \mathrm{w}_{\mathrm{i}-1}) = 1|$

and β and γ are constants.

Bigram Probability Estamate $=$

$$Pone(w_i\,|w_{i-1}) \quad = \frac{C\left(w_{i-1}^{i}\right)+\alpha\ Pone(w_i\,)}{C(w_{i-1})+\alpha} \tag{9}$$

$Where\ \alpha = \gamma[\mathrm{n}_1(\mathrm{w}_{\mathrm{i}-1}) + \beta]$ and $\mathrm{n}_1(\mathrm{w}_{\mathrm{i}-1}) = |\mathrm{w}_{\mathrm{i}}: \mathrm{C}(\mathrm{w}_{\mathrm{i}-1}) = 1|$

and β and γ are constants.

Unigram Probability Estamate $=$

$$Pone(w_i\,) \quad = \frac{C(w_i\,)}{Number\ of\ Tokens} \tag{10}$$

And if a word which does not exist in the unigram is assigned frequency 1.

Following tables represents estimated unigram bigram and trigram probabilities of sample sentence of section 4.5.1

"جب زلزله شروع ہوا اس وقت میں دفتر میں ہی تھا"

.

Pone estimated unigram probabilities of unigram words for above sample sentence is

Unigram Estimated Probability	Word Unigram
0.002246	جب
0.000236	زلزله
0.000815	شروع
0.00129	ہوا
0.012084	اس
0.001281	وقت
0.027346	میں
7.92E-05	دفتر
0.027346	میں
0.002313	ہی
0.002936	تھا

TABLE 4-16: EXAMPLE OF UNIGRAM WORDS AND THEIR ESTIMATED UNIGRAM PROBABILITES FOR SAMPLE SENTENCE

Pone estimated bigram probabilities of bigram words for the sample sentence in the word corpora is

Bigram Estimated Probabilities	Word Bigram
0.00396	جب <S>
0.000247	جب زلزله
0.000814	زلزله شروع
0.001664	شروع ہوا
0.012108	ہوا اس
0.006119	اس وقت
0.027461	وقت میں

7.43E-05	میں دفتر
0.027565	دفترمیں
0.002538	میں ہی
0.002979	ہی تھا

TABLE 4-17: EXAMPLE OF BIGRAM WORDS AND THEIR ESTIMATED BIGRAM PROBABILITES FOR SAMPLE SENTENCE

Pone estimated Trigram probabilities for the trigram words in the word corpora are given below in table 4-18.

Trigram Estimated Probabilities	Word Trigram
0.005283	؁؁جب
0.000247	؁جب زلزلہ
0.000814	جب زلزلہ شروع
0.001665	زلزلہ شروع ہوا
0.01211	شروع ہوا اس
0.006123	ہوا اس وقت
0.027429	اس وقت میں
7.45E-05	وقت میں دفتر
0.027565	میں دفترمیں
0.002538	دفترمیں ہی
0.002979	میں ہی تھا

TABLE 4-18: EXAMPLE OF TRIGRAM WORDS AND THEIR ESTIMATED TRIGRAM PROBABILITES FOR SAMPLE SENTENCE

5. GENERATING WORDS SEQUENCES

In this part of processing input is given in the form of ligatures separated with spaces. The function of this module is to get all possible word segments from the input ligatures and rank them. This process of generating the word sequences works in the building a tree like manner. First ligature is added as a root of tree and at each level of the tree maximum three or minimum two child nodes are added to each node. For example the second level of the tree contained following tree nodes

- The first node is composed of parent (root) string or next ligature combined with space.
- The Second node is composed of parent (root) string or next ligature combined without space.
- The third node is composed of parent (root) string or next ligature combined with zero-width-non-joiner if the ligature string of the parent node ends with a non joiner. Otherwise this node is not required and does not added in the current level of tree.

For example we have a sequence of three ligatures as"سو ئی گیس" Now the sequence of words are generated as follows

Tree Level	Input Ligature	Node String					
1	سو	سو					
2	ئی	سوئی			سو۔ئی		
3	گیس	سو ئیگیس	سوئی۔ گیس	سو ئی گیس	سو۔ ئیگیس	سو۔ئی ۔گیس	سو۔ ئی گیس

TABLE 5-1: PROCESS OF GENRATION OF WORDS SEQUENCE FROM LIGATURES SEQUENCE IN TREE MANNER

At each level, count is assigned to each node string. For assigning these counts firstly, all the space separated words are obtained from the node string. For each word of the node string, if this word exists in the dictionary then a count value is assigned to this word. This count value is equal to square of number of ligature this word is composed of. Otherwise if this word does not exist in dictionary then its count value is zero. The value of the node string is the sum of the word count of its words separated by space.

If a node string has only one word and if this word is not contained by dictionary as a valid word then it is also checked that this word may occur at the start of any dictionary entry. In this case word count is also assigned.

After assignment of word counts at each level, node strings are ranked according to these counts and best k (beam value) node strings are selected. These selected nodes are further explored for processing. The remaining lower ranked nodes and their respected strings are ignored in the processing in the next level. For example let say we have beam value k=3 for the above example

Tree Level	Input Ligature	Node String					
1	سو	1=سو					
2	ئی	4= سوئی			2= سو۔ ئی		
3	گیس	سوئیگیس =0	سوئی۔ گیس =5	سوئی گیس =9	سو۔ ئیگیس =1	سو۔ ئی ۔گیس =3	سو۔ ئی گیس =0

TABLE 5-2: ASSIGNING WORD COUNTS IN THE GENERATION PROCESS OF WORDS SEQUENCE FROM SAMPLE LIGATURE SEQUENCE WITH K=3

In the level 3 of the tree, three node strings سو ۔ ئی ۔گیس,سوئی ۔ گیس ,سوئی گیس are selected with valid word count 9, 5, 3 for the further processing.

Another example for the selection of the five best segments for the beam value =5 is as follows in table 5-3. In table 5-3 and in the subsequent tables '-' represents space character.

میں۔آ۔ج۔کل۔ا۔سی۔ا۔متحا۔ن۔۔کے۔لیے۔تیا۔ر۔ی۔کر۔ر۔ہا۔ہو۔ں	
Word Count	Resultant Segmentation Sequences
36	میں۔آج۔کل۔اسی۔امتحان۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں
36	میں۔آ۔ج۔کلاسی۔امتحان۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں
37	میں۔آ۔ج۔کل۔اسی۔امتحان۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں
35	میں۔آ۔ج۔کل۔اسی۔امتحان۔کے۔لیے۔تیاری۔کر۔ر۔ہا۔ہوں
36	میں۔آ۔ج۔کل اسی۔امتحان۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں

TABLE 5-3: SELECTION OF THE FIVE BEST SEGMENTS FOR THE SAMPLE SENTENCE ON THE BASIS OF VALID WORD COUNT

6. SELECTION OF THE BEST WORD SEGMENTATION SEQUENCE

For selection of the most probable word segmentation sequence, firstly all the word sequences with highest probabilities are found using all techniques presented in next sections. Then only one word sequence is selected which is the most occurring in the output of these techniques.

These techniques are variations of the Word Bigram Ligature Bigram technique. Derivation of Word Bigram Ligature Bigram is stated in Section 6.1 while its variations are presented in the succeeding sections as follows

6.1. LIGATURE BIGRAM AND WORD BIGRAM BASED TECHNIQUE

To derive equation for finding the maximum probable sequence of words among the k word sequences, obtained using valid word count heuristic, word language model is used. This language model is stated as

$$P(W) = \quad \text{argmax}_{w_1^n \in S} P(w_1^n) \tag{11}$$

Equation 11 represents a word sequence having a maximum probability where w_1^n represents a word sequence as $w_1^n = w_1, w_2, w_3, w_4, \_\_\_ w_n$ and S = set of the k maximum ranked word sequences. So Equation 11 can be written as

$$P(W) = \quad \text{argmax}_{w_1^n \in S} P(w_1, w_2, w_3, w_4, \_\_\_ w_n) \tag{12}$$

We can use the chain rule of probability to decompose the probability $P(w_1, w_2, w_3, w_4, \_\_\_ w_n)$ as

$$P(w_1, w_2, w_3, w_4, \_\_\_ w_n) = P(w_1)\, P(w_2|w_1)\, P(w_3|w_1^2) \ldots P(w_n|w_1^{n-1}) \tag{13}$$

$$P(w_1, w_2, w_3, w_4, \_\_\_ w_n) = \prod_1^n P(w_k|w_1^k) \tag{14}$$

To reduce the complexity of computing the w_1^{n-1} we will take the bigram model approximation in which probability of occurrence of a given word depends on its previous word, not all the previous words [27]. So equation (14) can be written as

$$P(w_1, w_2, w_3, w_4, \_\_\_ w_n) = \prod_1^n P(w_i|w_{i-1}) \tag{15}$$

In turn Equation (12) becomes

$$P(W) = \quad \text{argmax}_{w_1^n \in S} \prod_1^n P(w_i|w_{i-1}) \tag{16}$$

This equation gives us the most appropriate word sequences in a sentence or strings of words. But since we have ligature sequences as well so we can utilize relationship among these ligatures to make words So Equation (11) can be enhanced as

$$P(W) = \quad \text{argmax}_{w_1^n \in S} P(w_1^n | l_1^m) \tag{17}$$

Equation (17) gives a most probable sequence of words given a set S of word sequences w_1^n and a fix set of ligature sequence l_1^m where $w_1^n = w_{1,}w_{2,}w_{3,}w_{4,---}w_n$, $l_1^m = l_{1,}l_{2,}l_{3,}l_{4,---}l_m$: n represents number of words and m represents the number of ligatures. This equation also represent that m number of ligatures can be assigned to n number of words.

Now by applying the Bayesian theorem on equation (17)

$$P(w_1^n | L_1^m) = \frac{P(l_1^m | w_1^n).P(w_1^n)}{P(l_1^m)} \tag{18}$$

Putting the equation (18) in (17) we have

$$P(W) = \quad \text{argmax}_{w_1^n \in S} \frac{P(l_1^m | w_1^n).\, P(w_1^n)}{P(l_1^m)}$$

Where in Equation $P(l_1^m)$ remain constant for all w_1^n , So can be ignored as

$$P(W) = \quad \text{argmax}_{w_1^n \in S} P(l_1^m | w_1^n).\, P(w_1^n) \tag{19}$$

Where

$$P(l_1^m | w_1^n) = P\,(l_{1,}l_{2,}l_{3,}l_{4,---}l_m | w_1^n)$$

$$= P\,(l_1 | w_1^n) * P\,(l_2 | w_1^n l_1) * \; P(l_3 | w_1^n l_1 l_2) \; * P(l_4 | w_1^n l_1 l_2 l_3) * \ldots \; P(l_m | w_1^n l_1 l_2 l_3 \ldots l_{m-1})$$

Let's assume that a ligature l_i depends only on the word sequence w_1^n and its previous ligaturel_{i-1}, not all the previous ligature history so above equation can be written as

$$P(l_1^m | w_1^n) = P\,(l_1 | w_1^n) * P\,(l_2 | w_1^n l_1) * \; P(l_3 | w_1^n l_2) \; * P(l_4 | w_1^n l_3) * \ldots \; P(l_m | w_1^n l_{m-1})$$

$$= \prod_1^m P\,(l_i | w_1^n l_{i-1}) \tag{20}$$

Here we will take another assumption that l_i depends on the word in which it appears not whole word sequence. So (20) can be written as

$$P\left(l_i|w_1^n l_{i-1}\right) = P\left(l_i|w_k l_{i-1}\right) \quad (21)$$

Since we take assumption that l_i depends on w_k, a word in which l_i appears it gives always value of 1 and does not contribute in (20) So

$$P\left(l_i|w_1^n l_{i-1}\right) = P\left(l_i|l_{i-1}\right) \quad (22)$$

Now $P(w_1^n)$ of Equation (19) from [27] we have

$$P(w_1^n) = P(w_1) * P(w_2|w_1) * P(w_3|w_1^2) * \ldots * P(w_n|w_1^{n-1})$$

$$= \prod_{k=1}^{n} P\left(w_k|w_1^{k-1}\right) \quad (23)$$

Now using Markov assumption we assume that probability of a word depends only on the previous word which allows equation (24) to represented as

$$P(w_1^n) = \prod_{k=1}^{n} P(w_k|w_{k-1}) \quad (24)$$

Now putting values of (23) and (24) into (19) we have

$$P(W) = \operatorname{argmax}_{w_1^n \in S}\left(\prod_1^m (P\left(l_i|l_{i-1}\right)) * \left(\prod_{k=1}^{n} P(w_k|w_{k-1})\right)\right) \quad (25)$$

Equation (25) gives the maximum probable word sequence among the all alternative word sequences in set S.

Where

$P(w_k|w_{k-1})$and $P(l_i|l_{i-1})$ are estimated word bigram and ligature bigram probabilities calculated using equation (7) from word corpus and ligature corpus respectively.

Following table shows the probabilities of the five word sequences generated in the previous section using valid word count heuristic.

میں۔آج۔کل۔اسی۔امتحان۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں	
Ligature Bigram Word Bigram Probabilities	Resultant Segmentation Sequences
1.71517799431180 52E-71	میں۔آج۔کل۔اسی۔امتحان۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں

Ligature Bigram Word Bigram Probabilities	Resultant Segmentation Sequences
7.8096965343308147E-79	میں۔آ۔ج۔کلاسی۔امتحان۔ے۔لیے۔تیاری۔کر۔رہا۔ہوں
9.5099484655714152E-79	میں۔آ۔ج۔کل۔اسی۔امتحان۔ے۔لیے۔تیاری۔کر۔رہا۔ہوں
2.0751547770419163E-86	میں۔آ۔ج۔کل۔اسی۔امتحان۔ے۔لیے۔تیاری۔کر۔ر۔ہا۔ہوں
8.5004105827417516E-82	میں۔آ۔ج۔کل اسی۔امتحان۔ے۔لیے۔تیاری۔کر۔رہا۔ہوں۔

TABLE 6-1: PROBABILITIES OF THE FIVE WORD SEQUENCES USING LIGATURE BIGRAM WORD BIGRAM TECHNIQUE

6.2.LIGATURE BIGRAM BASED TECHNIQUE

On obtaining ranked valid sequences we can build the ligature bigram model by taking an assumption that sentences are made up of sequence of ligatures and space is also a valid ligature and it does not depends on the word history Then Equation (25) can be changed as

$$P(W) = \quad \text{argmax}_{w_1^n \in S}\left(\prod_1^m (P\left(l_i | l_{i-1}\right)\right) \qquad (26)$$

This model is based on the simplified assumption that a ligature depend on its previous ligature only and language model is independent of word's context. Here the Probability P is estimated Pone probability of l_{i-1} *and* l_i that occurs together in the corpus and these values are taken from the ligature bigram probabilities calculated before in Section 4.5.3.1. The ligature bigram probabilities for the ranked sentences in the above example is given as follows and best segmentation here is number 1 segment which has highest ligature bigram probability according to equation 26.

میں۔آ۔ج۔کل۔ا۔سی۔ا۔متحا۔ن۔ کے۔لیے۔تیا۔ر۔ی۔کر۔ر۔ہا۔ہو۔ں	
Ligature Bigram Probabilities	Resultant Segmentation Sequences
2.0380495084505667E-41	میں۔آج۔کل۔اسی۔امتحان۔ کے۔لیے۔تیاری۔کر۔رہا۔ہوں
2.0380495084505667E-41	میں۔آ۔ج۔کلاسی۔امتحان۔ کے۔لیے۔تیاری۔کر۔رہا۔ہوں
9.4368479194970159E-45	میں۔آ۔ج۔کل۔اسی۔امتحان۔ کے۔لیے۔تیاری۔کر۔رہا۔ہوں
3.9427588218834614E-46	میں۔آ۔ج۔کل۔اسی۔امتحان۔ کے۔لیے۔تیاری۔کر۔ر۔ہا۔ہوں
4.8154407914731122E-47	میں۔آ۔ج۔کل اسی۔امتحان۔ کے۔لیے۔تیاری۔کر۔رہا۔ہوں

TABLE 6-2: PROBABILITIES OF THE FIVE BEST RANKED WORD SEQUENCES USING LIGATURE BIGRAM TECHNIQUE

6.3.LIGATURE TRIGRAM BASED TECHNIQUE

Next variation of equation (25) is same as equation (26) except for this technique assumption is based on the ligature trigram model instead of ligature bigram that is a given ligature depends on its previous two ligatures. This variation is represented mathematically as

$$P(W) = \quad \text{argmax}_{w_1^n \in S}(\prod_1^m (P\ (l_i|l_{i-1}l_{i-2})) \qquad (27)$$

Here the Probability P is estimated Pone probability of the l_{i-2} , l_{i-1} and l_i occurs together in the corpus and these values are taken from the ligature trigram probabilities calculated before in section 4.5.3.1. In this ligature trigram model the best segmentation has highest ligature trigram probability. For the above example we have ligature trigram probabilities as follows in the table

میں۔آ۔ج۔کل۔ا۔سی۔ا۔متحا۔ن۔۔کے۔لیے۔تیا۔ر۔ی۔کر۔ر۔ہا۔ہو۔ں	
Ligature Trigram Probabilities	Resultant Segmentation Sequences
0.7172707063876166E-52	میں۔آج۔کل۔اسی۔امتحان۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں
3.4674482735904593E-55	میں۔آ۔ج۔کلاسی۔امتحان۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں
3.40199488419357E-57	میں۔آ۔ج۔کل۔اسی۔امتحان۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں
4.62493656794118E-60	میں۔آ۔ج۔کل۔اسی۔امتحان۔کے۔لیے۔تیاری۔کر۔ر۔ہا۔ہوں
3.7714500531151638E-58	میں۔آ۔ج۔کل اسی۔امتحان۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں

TABLE 6-3: PROBABILITIES OF THE FIVE BEST RANKED WORD SEQUENCES USING LIGATURE TRIGRAM TECHNIQUE

6.4.WORD BIGRAM BASED TECHNIQUE

In this technique Word Bigram model is used to decide the most appropriate segmentation among the list of candidate word segmented sequences. The main idea is to assume that the next word can be predicted given the previous word and the ligatures of the words are independent of each other. Therefore, the probability model of equation (25) can be changed as

$$P(W) = \quad \text{argmax}_{w_1^n \in S}(\prod_{k=1}^{n} P(w_k|w_{k-1}) \qquad (28)$$

Here the Probability $P(w_k|w_{k-1})$ is estimated Pone Probability calculated from word corpora. For example bigram probabilities for the following word sequences are as follows

میں۔آ۔ج۔کل۔ا۔سی۔ا۔متحا۔ن۔۔کے۔لیے۔تیا۔ر۔ی۔کر۔ر۔ہا۔ہو۔ں	
Word Bigram Probabilities	Resultant Segmentation Sequences
8.415781791364698E-31	میں۔آج۔کل۔اسی۔امتحان۔۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں
1.7652403904111454E-35	میں۔آ۔ج۔کلاسی۔امتحان۔۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں
1.0077462884533054E-34	میں۔آ۔ج۔کل۔اسی۔امتحان۔۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں
5.2632049556878856E-41	میں۔آ۔ج۔کل۔اسی۔امتحان۔۔کے۔لیے۔تیاری۔کر۔ر۔ہا۔ہوں
1.7652403904111454E-35	میں۔آ۔ج۔کل اسی۔امتحان۔۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں

TABLE 6-4: PROBABILITIES OF THE FIVE BEST RANKED WORD SEQUENCES USING WORD BIGRAM TECHNIQUE

6.5.WORD TRIGRAM BASED TECHNIQUE

This technique is similar to technique presented in the section 6.4. Only one variation we have in this technique is that we use the word trigram Markov assumption rather than word bigram Markov assumption [27] which changes the equation (25) as

$$P(W) = \quad \text{argmax}_{w_1^n \in S}\left(\prod_{k=1}^{n} P(w_k|w_{k-1}w_{k-2}\right) \qquad (29)$$

Here the Probability $P(w_k|w_{k-1}w_{k-2})$ is estimated and smoothed Pone word trigram probability calculated from word corpora. For example trigram probabilities for the following word sequences are

میں۔آ۔ج۔کل۔ا۔سی۔ا۔متحا۔ن۔۔کے۔لیے۔تیا۔ر۔ی۔کر۔ر۔ہا۔ہو۔ں	
Word Trigram Probabilities	Resultant Segmentation Sequences
9.7988201096147428E-31	میں۔آج۔کل۔اسی۔امتحان۔۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں
3.7912813389580995E-37	میں۔آ۔ج۔کلاسی۔امتحان۔۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں
1.3310574106509475E-35	میں۔آ۔ج۔کل۔اسی۔امتحان۔۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں
3.7162699250808728E-45	میں۔آ۔ج۔کل۔اسی۔امتحان۔۔کے۔لیے۔تیاری۔کر۔ر۔ہا۔ہوں
3.7912813389580995E-37	میں۔آ۔ج۔کل اسی۔امتحان۔۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں

TABLE 6-5 : PROBABILITIES OF THE FIVE BEST RANKED WORD SEQUENCES USING WORD TRIGRAM TECHNIQUE

6.6.LIGATURE TRIGRAM AND WORD BIGRAM BASED TECHNIQUE

In this technique equation (25) is changed with an assumption that a ligature depends on the previous two ligatures rather on previous one ligature. This assumption results in following change in the equation (25)

$$P(W) = \quad \text{argmax}_{w_1^n \in S}\left(\prod_1^m (P\left(l_i | l_{i-1} l_{i-2}\right)\right) * \left(\prod_{k=1}^n P(w_k | w_{k-1})\right) \qquad (30)$$

This equation (30) gives the maximum probable word sequence among all alternative word sequences in set S. Where $P(w_k|w_{k-1})$ is estimated and smooth word bigram probability and $P(l_i|l_{i-1}l_{i-2})$ is estimated ligature trigram probability.

میں۔آ۔ج۔کل۔ا۔سی۔ا۔متحا۔ن۔۔کے۔لیے۔تیا۔ر۔ی۔کر۔ر۔ہا۔ہو۔ں	
Ligature Trigram and Word Bigram Probabilities	Resultant Segmentation Sequences
2.2867957333025595E-82	میں۔آج۔کل۔اسی۔امتحان۔۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں
6.1208797442032741E-90	میں۔آ۔ج۔کلاسی۔امتحان۔۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں
3.428347717883203E-91	میں۔آ۔ج۔کل۔اسی۔امتحان۔۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں
2.4341989064130139E-100	میں۔آ۔ج۔کل۔اسی۔امتحان۔۔کے۔لیے۔تیاری۔کر۔ر۔ہا۔ہوں
6.6575159641771464E-93	میں۔آ۔ج۔کل اسی۔امتحان۔۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں

TABLE 6-6: PROBABILITIES OF THE FIVE BEST RANKED WORD SEQUENCES USING LIGATURE TRIGRAM AND WORD BIGRAM TECHNIQUE

6.7.LIGATURE BIGRAM AND WORD TRIGRAM BASED TECHNIQUE

Another variation can be done in Equation (25) with a supposition that a word depends on the previous two words in a text which results in following form of equation

$$P(W) = \quad \text{argmax}_{w_1^n \in S}\left(\prod_1^m (P\left(l_i | l_{i-1}\right)\right) * \left(\prod_{k=1}^n P(w_k | w_{k-1} w_{k-2})\right) \qquad (31)$$

This equation (31) gives the maximum probable word sequence among the all word sequences in set S. Where $P(w_k|w_{k-1}w_{k-2})$ probability value is obtained from the estimated Pone word trigram probability list and $P(l_i|l_{i-1})$ probability value is obtained from the estimated ligature bigram probability list.

میں۔آ۔ج۔کل۔ا۔سی۔ا۔متحا۔ن۔۔کے۔لیے۔تیا۔ر۔ی۔کر۔ر۔ہا۔ہو۔ں	
Ligature Bigram and Word Trigram Probabilities	Resultant Segmentation Sequences
1.9970480507795855E-71	میں۔آج۔کل۔اسی۔امتحان۔۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں
1.6773215078450551E-80	میں۔آ۔ج۔کلاسی۔امتحان۔۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں
1.2560986356432479E-79	میں۔آ۔ج۔کل۔اسی۔امتحان۔۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں
1.46523560316128E-90	میں۔آ۔ج۔کل۔اسی۔امتحان۔۔کے۔لیے۔تیاری۔کر۔ر۔ہا۔ہوں
1.8256690811569632E-83	میں۔آ۔ج۔کل اسی۔امتحان۔۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں

TABLE 6-7: PROBABILITIES OF THE FIVE BEST RANKED WORD SEQUENCES USING LIGATURE BIGRAM AND WORD TRIGRAM TECHNIQUE

6.8.LIGATURE TRIGRAM AND WORD TRIGRAM BASED TECHNIQUE

For the Next variation of Equation (25) we can suppose that a word depends on the previous two words in a text as well as, a ligature also depends on the previous two ligatures which results in following form of equation

$$P(W) = \quad \text{argmax}_{w_1^n \in S}\left(\prod_1^m (P\left(l_i | l_{i-1} l_{i-2}\right)\right) * \left(\prod_{k=1}^n P(w_k | w_{k-1} w_{k-2})\right) \qquad (32)$$

This equation (32) gives the maximum probable word sequence among all word sequences of set S. Where $P(w_k|w_{k-1}w_{k-2})$ probability value is obtained from the estimated Pone word trigram probability list and $P(l_i|l_{i-1}l_{i-2})$ probability value is obtained from the estimated ligature trigram probability list calculated from the corpus.

میں۔آج۔کل۔اسی۔امتحان۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں	
Ligature Trigram and Word Trigram Probabilities	Resultant Segmentation Sequences
2.6626046841018035E-82	میں۔آج۔کل۔اسی۔امتحان۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں
1.3146071933465986E-91	میں۔آج۔کلاسی۔امتحان۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں
4.5282505016024634E-92	میں۔آج۔کل۔اسی۔امتحان۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں
1.7187512672846559E-104	میں۔آج۔کل۔اسی۔امتحان۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں
1.4298628207188055E-94	میں۔آج۔کل اسی۔امتحان۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں

TABLE 6-8: PROBABILITIES OF THE FIVE BEST RANKED WORD SEQUENCES USING LIGATURE TRIGRAM AND WORD TRIGRAM TECHNIQUE

6.9.NORMALIZED LIGATURE BIGRAM AND WORD BIGRAM BASED TECHNIQUE

Normalized ligature gram and word gram based technique is similar to the ligature bigram and word bigram based techniques. Instead in this technique ligature bigram and word bigram are normalized using nth root formula. This normalization is done through number of ligature grams exist in the corpus and number of word grams exists in the corpus. This changes the equation (25)as follows

$$P(W) = \operatorname{argmax}_{w_1^n \in S} \left(\prod_{1}^{m} (P\ (l_i | l_{i-1}) \right)^{1/NL} * \left(\prod_{k=1}^{n} P(w_k | w_{k-1}) \right)^{1/NW} \quad (33)$$

This equation (33) gives the maximum probable word sequence. Where $P(w_k|w_{k-1})$ probability value is obtained from the estimated Pone word bigram probability list and $P(l_i|l_{i-1})$ probability value is obtained from the estimated ligature bigram probability list calculated from the corpus.

NL represents the number of ligature bigrams exit in the corpus and NW represents the number of word bigram exists in the corpus. Following technique shows the probabilities of the five best ranked sentences calculated using equation (33).

میں۔آ۔ج۔کل۔ا۔سی۔ا۔متحا۔ن۔کے۔لیے۔تیا۔ر۔ی۔کر۔ر۔ہا۔ہو۔ں	
Normalized Ligature Bigram and Word Bigram Probabilities	Resultant Segmentation Sequences
0.000072910009059227 67	میں۔آج۔کل۔اسی۔امتحان۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں
0.000034105449123147042	میں۔آ۔ج۔کلاسی۔امتحان۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں
0.00005005415865115705	میں۔آ۔ج۔کل۔اسی۔امتحان۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں
0.000027351618546819751	میں۔آ۔ج۔کل۔اسی۔امتحان۔کے۔لیے۔تیاری۔کر۔ر۔ہا۔ہوں
0.000030502974525510 13	میں۔آ۔ج۔کل اسی۔امتحان۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں

TABLE 6-9: PROBABILITIES OF THE FIVE BEST RANKED WORD SEQUENCES USING NORMALIZED LIGATURE BIGRAM AND WORD BIGRAM TECHNIQUE

6.10. NORMALIZED LIGATURE TRIGRAM AND WORD BIGRAM BASED TECHNIQUE

Like the previous technique this technique is ligature trigram and word bigram based techniques. Difference lies only in normalization of the values. After normalization equation (30) changes as

$$P(W) = \quad \text{argmax}_{w_1^n \in S}\left(\prod_{1}^{m}(P\left(l_i|l_{i-1}l_{i-2}\right)^{1/NL}\right) * \left(\prod_{k=1}^{n} P(w_k|w_{k-1})\right)^{1/NW} \qquad (34)$$

In this equation (34)$P(w_k|w_{k-1})$ probability value is obtained from the estimated Pone word bigram probability list and $P(l_i|l_{i-1}l_{i-2})$ probability value is obtained from the estimated ligature trigram probability list calculated from the corpus.NL represents the number of ligature trigrams exit in the corpus and NW represents the number of word bigram exists in the corpus. Following technique shows the probabilities of the five best ranked sentences calculated using equation (34).

میں۔آج۔کل۔اسی۔امتحان۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں	
Ligature Trigram and Word Bigram Probabilities	Resultant Segmentation Sequences
0.000026563149607630293	میں۔آج۔کل۔اسی۔امتحان۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں
0.0000090122926195169864	میں۔آج۔کلاسی۔امتحان۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں
0.000016586865989870657	میں۔آج۔کل۔اسی۔امتحان۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں
0.000007171397606986245	میں۔آج۔کل۔اسی۔امتحان۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں

0.00000085101640000236287	میں۔آ۔ج۔کل۔اسی۔امتحان۔۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں

TABLE 6-10 : PROBABILITIES OF THE FIVE BEST RANKED WORD SEQUENCES USING NORMALIZED LIGATURE TRIGRAM AND WORD BIGRAM TECHNIQUE

6.11. NORMALIZED LIGATURE BIGRAM AND WORD TRIGRAM BASED TECHNIQUE

This technique is Similar to equation (31) and uses normalized values. This technique is mathematically represented as

$$P(W) = \operatorname{argmax}_{w_1^n \in S}\left(\prod_{1}^{m}(P\,(l_i|l_{i-1})\right)^{1/NL} * \left(\prod_{k=1}^{n} P(w_k|w_{k-1}w_{k-2})\right)^{1/NW} \quad (35)$$

This equation (35) gives the maximum probable word sequence. Where $P(w_k|w_{k-1}w_{k-2})$ probability value is obtained from the estimated Pone word trigram probability list and $P(l_i|l_{i-1})$ probability value is obtained from the estimated ligature bigram probability list calculated from the corpus.NL represents the number of ligature bigrams exit in the corpus and NW represents the number of word trigrams exist in the corpus. Following technique shows the probabilities of the five best ranked sentences calculated using equation (35).

میں۔آ۔ج۔کل۔ا۔سی۔ا۔متحا۔ن۔۔کے۔لیے۔تیا۔ر۔ی۔کر۔ر۔ہا۔ہو۔ں	
Normalized Ligature Bigram and Word Trigram Probabilities	Resultant Segmentation Sequences
0.00007392551572460787	میں۔آج۔کل۔اسی۔امتحان۔۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں

0.00000069444565163656789	میں۔آ۔ج۔کلاسی۔امتحان۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں
0.000023012651843930476	میں۔آ۔ج۔کل۔اسی۔امتحان۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں
0.000001236969863620304 3	میں۔آ۔ج۔کل۔اسی۔امتحان۔کے۔لیے۔تیاری۔کر۔ر۔ہا۔ہوں
0.000000621093067701168 18	میں۔آ۔ج۔کل اسی۔امتحان۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں

TABLE 6-11: PROBABILITIES OF THE FIVE BEST RANKED WORD SEQUENCES USING NORMALIZED LIGATURE BIGRAM AND WORD TRIGRAM TECHNIQUE

6.12. NORMALIZED LIGATURE TRIGRAM AND WORD TRIGRAM BASED TECHNIQUE

This technique is Similar to equation (32) but uses normalized values. This technique is mathematically represented as

$$P(W) = \operatorname{argmax}_{w_1^n \in S}\left(\prod_{1}^{m}(P\,(l_i|l_{i-1}l_{i-2})\right)^{1/NL} * \left(\prod_{k=1}^{n} P(w_k|w_{k-1}w_{k-2})\right)^{1/NW} \quad (36)$$

This equation (36) gives the maximum probable word sequence. Where $P(w_k|w_{k-1}w_{k-2})$ probability value is obtained from the estimated Pone word trigram probability list and $P(l_i|l_{i-1}l_{i-2})$ probability value is obtained from the estimated ligature trigram probability list calculated from the corpus.NL represents the number of ligature trigrams exit in the corpus and NW represents the number of word trigrams exist in the corpus. Following technique shows the probabilities of the five best ranked sentences calculated using equation (36).

میں۔آ۔ج۔کل۔ا۔سی۔ا۔متحا۔ن۔۔کے۔لیے۔تیا۔ر۔ی۔کر۔ر۔ہا۔ہو۔ں	
Normalized Ligature Trigram and Word Trigram Probabilities	Resultant Segmentation Sequences
0.000026933127006180978	میں۔آج۔کل۔اسی۔امتحان۔۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں
0.00000185758481102578 16	میں۔آ۔ج۔کلاسی۔امتحان۔۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں
0.0000076258952800918	میں۔آ۔ج۔کل۔اسی۔امتحان۔۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں
0.00000324324599098073 65	میں۔آ۔ج۔کل۔اسی۔امتحان۔۔کے۔لیے۔تیاری۔کر۔ر۔ہا۔ہوں
0.00000173281588029858 64	میں۔آ۔ج۔کل اسی۔امتحان۔۔کے۔لیے۔تیاری۔کر۔رہا۔ہوں

TABLE *6-12:* PROBABILITIES OF THE FIVE BEST RANKED WORD SEQUENCES USING NORMALIZED LIGATURE TRIGRAM AND WORD TRIGRAM TECHNIQUE

7. RESULTS AND DICUSSION

The algorithm was tested on a corpus of 150 sentences composed of 2156 words and 6075 ligatures. In these sentences, 62 words are unknown and 2092 are known words. Unknown words mean here, the words that do not exist in our dictionary. The average length of the sentence is 14 in terms of words and 40.5 in terms of ligatures. The average length of the word is 2.81 in terms of ligatures. At the start we have tested all the techniques presented in section 6 for the beam value of 10, 20,30,40,50.

The Results for the Ligature Bigram technique are as follows

Beam Value	Total Sentences identified	%age	Total Words Identified	%age	Total known words identified	%age	Total unknown words identified	%age
10	81/150	54%	1978/2156	91.74%	1946/2092	93.02%	32/64	50%
20	65/150	43.33%	1914/2156	88.78%	1882/2092	89.96%	32/64	50%
30	59/150	39.33%	1895/2156	87.89%	1859/2092	88.86%	36/64	56.25%
40	54/150	36%	1854/2156	85.99%	1825/2092	87.24%	29/64	45.31%
50	50/150	33.33%	1835/2156	85.11%	1806/2092	86.33%	29/64	45.31%

TABLE 7-1: RESULTS FOR THE LIGATURE BIGRAM TECHNIQUE

The Results for the Ligature Trigram technique are as follows

Beam Value	Total Sentences identified	%age	Total Words Identified	%age	Total known words identified	%age	Total unknown words identified	%age
10	45/150	30%	1848/2156	85.71%	1817/2092	86.86%	31/64	48.44%
20	35/150	23.33%	1776/2156	82.38%	1745/2092	83.41%	31/64	48.44%
30	30/150	20%	1723/2156	79.92%	1691/2092	80.83%	32/64	50%
40	22/150	14.67%	1689/2156	78.34%	1661/2092	79.40%	28/64	43.75%
50	16/150	10.67%	1637/2156	75.93%	1610/2092	76.96%	27/64	42.19%

TABLE 7-2: RESULTS FOR THE LIGATURE TRIGRAM TECHNIQUE

The Results for the Word Bigram technique are as follows

Beam Value	Total Sentences identified	%age	Total Words Identified	%age	Total known words identified	%age	Total unknown words identified	%age
10	93/150	62%	2015/2156	93.46%	1981/2092	94.69%	34/64	53.13%
20	72/150	48%	1936/2156	89.80%	1900/2092	90.82%	36/64	56.25%
30	65/150	43.33%	1903/2156	88.27%	1866/2092	89.20%	37/64	57.81%
40	58/150	38.67%	1862/2156	86.36%	1829/2092	87.43%	33/64	51.56%
50	47/150	31.33%	1827/2156	84.74%	1796/2092	85.85%	31/64	48.44%

TABLE 7-3: RESULTS FOR THE WORD BIGRAM TECHNIQUE

The Results for the Word Trigram technique are as follows

Beam Value	Total Sentences identified	%age	Total Words Identified	%age	Total known words identified	%age	Total unknown words identified	%age
10	98/150	65.33%	2029/2156	94.11%	1995/2092	95.36%	34/64	53.13%
20	93/150	62%	2008/2156	93.14%	1971/2092	94.22%	37/64	57.81%
30	88/150	58.67%	1995/2156	92.53%	1957/2092	93.55%	38/64	59.38%
40	80/150	53.33%	1955/2156	90.68%	1921/2092	91.83%	34/64	53.13%
50	74/150	49.33%	1937/2156	89.84%	1903/2092	90.97%	34/64	53.13%

TABLE 7-4: RESULTS FOR THE WORD TRIGRAM TECHNIQUE

From the Table 7-1, Table 7-2, Table 7-3, Table 7-4 it can be analyzed that the statistical word bigram and word trigram techniques clearly outperforms then ligature bigram and ligature trigram techniques The reason behind this is , the difference in amount of corpora used for calculation of ligature grams and word grams. As corpora used for the ligature grams is composed of half million words while the corpora used for the word grams is composed of 18 million words. So there is huge difference in amount of corpora and effect of these corpora can be viewed by the hit ratio. For example the hit ratio for the ligature trigram is percentage of the ligature trigrams which exist in the ligature corpora. Table 7-5 shows the hit ratios of the ligature grams and word grams in the ligature and word corpora respectively.

Technique Name	Ligature Bigram	Ligature Trigram	Word Bigram	Word Trigram
Hit Ratio (%)	98%	71%	96 %	88%

TABLE 7-5: HIT RATIOS OF THE LIGATURE GRAMS AND WORD GRAMS

So the results from ligature bigram and ligature trigram techniques are expected to improve a lot once the amount of corpora is increased. These results also affect the other subsequent techniques to improve the results.

The Results for the Ligature Bigram and Word Bigram technique are as follows

Beam Value	Total Sentences identified	%age	Total Words Identified	%age	Total known words identified	%age	Total unknown words identified	%age
10	98/150	65.33%	2027/2156	94.02%	1992/2092	95.22%	35/64	54.69%
20	82/150	54.67%	1969/2156	91.33%	1932/2092	92.35%	37/64	57.81%
30	76/150	50.67%	1945/2156	90.21%	1907/2092	91.16%	38/64	59.38%
40	72/150	48%	1909/2156	88.54%	1876/2092	89.68%	33/64	51.56%
50	68/150	45.33%	1900/2156	88.13%	1865/2092	89.15%	35/64	54.69%

TABLE 7-6: RESULTS FOR THE LIGATURE BIGRAM AND WORD BIGRAM

The Results for the Ligature Bigram and Word Trigram technique are as follows

Beam Value	Total Sentences identified	%age	Total Words Identified	%age	Total known words identified	%age	Total unknown words identified	%age
10	101/150	67.33%	2043/2156	94.76%	2010/2092	96.08%	33/64	51.56%
20	96/150	64%	2018/2156	93.60%	1982/2092	94.74%	36/64	56.25%
30	93/150	62%	2007/2156	93.09%	1969/2092	94.12%	38/64	59.38%
40	84/150	56%	1964/2156	91.10%	1931/2092	92.30%	33/64	51.56%
50	83/150	55.33%	1960/2156	90.91%	1924/2092	91.97%	36/64	56.25%

TABLE 7-7: RESULTS FOR THE LIGATURE BIGRAM AND WORD TRIGRAM

The Results for the Ligature Trigram and Word Bigram technique are as follows

Beam Value	Total Sentences identified	%age	Total Words Identified	%age	Total known words identified	%age	Total unknown words identified	%age
10	72/150	48%	1946/2156	90.26%	1912/2092	91.40%	34/64	53.13%
20	58/150	38.67%	1865/2156	86.50%	1832/2092	87.57%	33/64	51.56%
30	50/150	33.33%	1827/2156	84.74%	1792/2092	85.66%	35/64	54.69%
40	46/150	30.67%	1795/2156	83.26%	1766/2092	84.42%	29/64	45.31%
50	42/150	28%	1776/2156	82.38%	1746/2092	83.46%	30/64	46.88%

TABLE 7-8: RESULTS FOR THE LIGATURE TRIGRAM AND WORD BIGRAM TECHNIQUE

The Results for the Ligature Trigram and Word Trigram technique are as follows

Beam Value	Total Sentences identified	%age	Total Words Identified	%age	Total known words identified	%age	Total unknown words identified	%age
10	94/150	62.67%	2016/2156	93.51%	1983/2092	94.79%	33/64	51.56%
20	58/150	38.67%	1865/2156	86.50%	1832/2092	87.57%	33/64	51.56%
30	70/150	46.67%	1920/2156	89.05%	1886/2092	90.15%	34/64	53.13%
40	64/150	42.67%	1878/2156	87.11%	1849/2092	88.38%	29/64	45.31%
50	62/150	41.33%	1868/2156	86.64%	1835/2092	87.72%	33/64	51.56%

TABLE 7-9: RESULTS FOR THE LIGATURE TRIGRAM AND WORD TRIGRAM

The Results for the Normalized Ligature Bigram and Word Bigram technique are as follows

Beam Value	Total Sentences identified	%age	Total Words Identified	%age	Total known words identified	%age	Total unknown words identified	%age
10	87/150	58%	2044/2156	94.81%	2007/2092	95.94%	37/64	57.81%
20	87/150	58%	2049/2156	95.04%	2009/2092	96.03%	40/64	62.50%
30	87/150	58%	2056/2156	95.36%	2013/2092	96.22%	43/64	67.19%
40	89/150	59.33%	2058/2156	95.46%	2016/2092	96.37%	42/64	65.63%
50	90/150	60%	2067/2156	95.87%	2024/2092	96.75%	43/64	67.19%

TABLE 7-10: RESULTS FOR THE NORMALIZED LIGATURE BIGRAM AND WORD BIGRAM TECHNIQUE

The Results for the Normalized Ligature Bigram and Word Trigram technique are as follows

Beam Value	Total Sentences identified	%age	Total Words Identified	%age	Total known words identified	%age	Total unknown words identified	%age
10	95/150	63.33%	2048/2156	94.99%	2012/2092	96.18%	36/64	56.25%
20	97/150	64.67%	2059/2156	95.50%	2018/2092	96.46%	41/64	64.06%
30	98/150	65.33%	2064/2156	95.73%	2022/2092	96.65%	42/64	65.63%
40	99/150	66%	2065/2156	95.78%	2024/2092	96.75%	41/64	64.06%
50	100/150	66.67%	2070/2156	96.01%	2028/2092	96.94%	42/64	65.63%

TABLE 7-11: RESULTS FOR THE NORMALIZED LIGATURE BIGRAM AND WORD TRIGRAM TECHNIQUE

The Results for the Normalized Ligature Trigram and Word Bigram technique are as follows

Beam Value	Total Sentences identified	%age	Total Words Identified	%age	Total known words identified	%age	Total unknown words identified	%age
10	90/150	60%	2049/2156	95.04%	2012/2092	96.18%	37/64	57.81%
20	97/150	64.67%	2059/2156	95.50%	2018/2092	96.46%	41/64	64.06%
30	90/150	60%	2059/2156	95.50%	2018/2092	96.46%	41/64	64.06%
40	92/150	61.33%	2061/2156	95.59%	2021/2092	96.61%	40/64	62.50%
50	93/150	62%	2071/2156	96.06%	2030/2092	97.04%	41/64	64.06%

TABLE 7-12: RESULTS FOR THE NORMALIZED LIGATURE TRIGRAM AND WORD BIGRAM TECHNIQUE

The Results for the Normalized Ligature Trigram and Word Trigram technique are as follows

Beam Value	Total Sentences identified	%age	Total Words Identified	%age	Total known words identified	%age	Total unknown words identified	%age
10	95/150	63.33%	2048/2156	94.99%	2012/2092	96.18%	36/64	56.25%
20	97/150	64.67%	2059/2156	95.50%	2018/2092	96.46%	41/64	64.06%
30	99/150	66%	2068/2156	95.92%	2026/2092	96.85%	42/64	65.63%
40	100/150	66.67%	2067/2156	95.87%	2026/2092	96.85%	41/64	64.06%
50	101/150	67.33%	2072/2156	96.10%	2030/2092	97.04%	42/64	65.63%

TABLE 7-13: NORMALIZED LIGATURE TRIGRAM AND WORD TRIGRAM TECHNIQUE

The Results for the optimal technique on the vote basis of all the 12 techniques are as follows

Beam Value	Total Sentences identified	%age	Total Words Identified	%age	Total known words identified	%age	Total unknown words identified	%age
10	105/150	70%	2051/2156	95.13%	2016/2092	96.37%	35/64	54.69%
20	99/150	66%	2021/2156	93.74%	1983/2092	94.79%	38/64	59.38%
30	101/150	67.33%	2031/2156	94.20%	1991/2092	95.17%	40/64	62.50%
40	89/150	59.33%	1983/2156	91.98%	1947/2092	93.07%	36/64	56.25%
50	91/150	60.67%	1987/2156	92.16%	1948/2092	93.12%	39/64	60.94%

TABLE 7-14: RESULTS FOR THE OPTIMAL TECHNIQUE ON THE VOTE BASIS OF ALL THE 12 TECHNIQUES

As it can be observed from the above tables that following two techniques

1. Ligature Trigram based Technique
2. Ligature Trigram and word bigram based technique

Behaved adversely in the identification of sentences and it also vote falsely for the selection of the optimal word sequences and do not contribute in the selection of the optimal solution So these techniques are excluded to vote for the most favorable solution. The results for the optimal technique after the exclusion of these techniques for the beam values 10, 20, 30, 40 and 50 are as follows.

Beam Value	Total Sentences identified	%age	Total Words Identified	%age	Total known words identified	%age	Total unknown words identified	%age
10	110/150	73.33%	2060/2156	95.55%	2024/2092	96.75%	36/64	56.25%

20	112/150	74.67%	2066/2156	95.83%	2027/2092	96.89%	39/64	60.94%
30	114/150	76%	2062/2156	95.64%	2019/2083	96.93%	43/73	58.90%
40	105/150	70%	2037/2156	94.48%	2000/2092	95.60%	37/64	57.81%
50	106/150	70.67%	2040/2156	94.62%	2000/2092	95.60%	40/64	62.50%

TABLE 7-15: RESULTS FOR THE OPTIMAL TECHNIQUE ON THE VOTE BASIS OF ALL THE 10 TECHNIQUES

This table shows that system performs better on the beam value of 20 from the other beam values. Therefore beam value 20 is selected for the word segmentation model.

Three types of errors are considered here First type of errors are Sentence identification errors, second type of errors are Known word recognition errors and third type of errors are Unknown words recognition errors.

First type of errors is sentence identification errors. A sentence is considered incorrect even if one word of the sentence is identified wrongly. This type of errors depends on the other two types of errors. For example for the beam value of 20 we have 38 sentences incorrect. In the 38 sentences 25 sentences are identified in the wrong way due to unknown words errors and remaining 13 errors are due to known word identification errors. So improvement in recognition of other two types of errors results in the improvement of sentence identification rate.

Second type of errors is known words identification errors. Most of the errors in this category are of space insertion means two words are joined together and space is deleted from them. The reason of these errors is insufficient cleaning of word grams as discussed in section 4.5.2.1. The words with frequency greater than 50 in the unigram list, which covers 18962196 words, are find out and cleaned. Other low frequency words cause these errors for example errors "بنیادپر", "سےتقسیم" are

space insertion errors and these error words exits in word corpora with frequency 40 and 5 respectively which falsify our results. There are 14 errors of space insertion which results in the 47

known words recognition errors as one space insertion error result in two or more known words recognition errors for example "بنیادپر" is a space insertion error which results in two known word recognition errors. If low frequency words are also cleaned from the word grams lists then error rate for the space insertion errors will become low and results of known word recognition errors will definitely improved. Other errors in this category are due to incorrect selection of beam value.

Third type of errors is unknown word recognition errors. These words do not exist in the dictionary. Most of these errors are recognized as real word errors. Real word spelling is words in a text that, although correctly spelled words in the dictionary, are not the words that the writer intended. For example a word "کارتک" is a proper noun and does not exist in dictionary. This system recognizes it as two words "کار تک" which are valid words of dictionary. Other unknown words which are incorrectly identified are diacritize words. So the unknown words rate can be further improved by enhancing dictionary with diacritize words along with the words without diacritics.

8. CONCLUSION

This theses work presents a starting effort on statistical solution of word segmentation problem for Urdu OCR systems and simultaneously for the Urdu language. In other south Asian languages, like Chinese, have only space insertion problem. Here the Urdu language differs from these languages as it also face space removal and zero-width- non joiner insertion problems with the space insertion problem. All these problems have their own dimension and require intensive cleaned data. This work tries to solve all these problems and effectively solve space removal problems but space insertion problem require more detailed analysis and cleaning.

Ligature grams results are poor than word grams techniques, for the effectiveness of the ligature gram techniques huge amount of cleaned data for ligature grams is required.

9. FUTURE WORK AND IMPROVEMENTS

This thesis work used the knowledge of ligature grams and word grams. This work can be further enhanced by using the character grams information. In this work Statistics are only used for the word segmentation so the Urdu Rules for the formation of words or rule based techniques can also be used along with the statistics information to improve the results.

We have tried to clean the corpus with respect to space removal, space insertion and ZWNJ insertion. These lists are need to be improved as well as abbreviations and English words are needed to handle more effectively.

The unknown word detection rate can be increased efficiently by applying POS tagging techniques or word net based techniques with the minimum distance which results in the improvement of the real word detection errors.

Other issues are related to memory as the loading of the word trigram requires huge memory. This problem can be handled by reducing the amount of trigrams by using some grammatical trigram techniques.

Reference:

[1] <http://en.wikipedia.org/wiki/Word>

[2] Zaheer Ahmad, Jehanzeb Khan Orakzai, Inam Shamsher, and Awais Adnan, "*Urdu Nastaleeq Optical Character Recognition*", Proceedings of World Academy of Science, Engineering And Technology, Volume 26, December 2007

[3] U. Pal and Anirban Sarkar, "*Recognition of Printed Urdu Script*", Pro. Seventh International Conference on Document Analysis and Recognition, pp 1183-1187, 2003

[4] Poowarawan, Y., "*Dictionary-based Thai Syllable Separation*", Proceedings of the Ninth Electronics Engineering Conference, 1986

[5] Fung Pascale and Wu Dekai, "*Statistical augmentation of a Chinese machine readable dictionary*", 1994

[6] Alexander Clark1 and Shalom Lappin2, "*Grammar Induction Through Machine Learning*".

[7] Theeramunkong, T., Sornlertlamvanich, V., Tanhermhong, T., Chinnan, W., "*Character-Cluster Based Thai Information Retrieval*", Proceedings of the Fifth International Workshop on Information Retrieval with Asian Languages, September 30 - October 20, 2000, Hong Kong, pp.75-80.

[8] Quinlan, J.R., "*Induction of Decision Trees, Machine Learning*", 1, pp. 81-106, 1986.

[9] Charoenporn. T., Sornlertlamvanich. V. and Isaraha. H. 1997. "*Building A Large Thai Text Corpus-Part-Of-Speech Tagged Corpus:ORCHID*". NECTEC, Bangkok.

[10] Richard Sproat, Chilin Shih, William Gale and Nancy Chang (1996), "*A Stochastic Finite-State Word-Segmentation Algorithm for Chinese*", Computational Linguistics, Vol 22, Number 3, 1996

[11] Chang, Jyun-Shen, Shun-De Chen, Ying Zhen, Xian-Zhong Liu and Shu-Jin Ke, "*Large-corpus-based methods for Chinese personal name recognition*", Journal of Chinese Information Processing, 6(3):7-15 , 1992

[12] Li Haizhou et al, "*Pinyin Streamer: Chinese pinyin to hanzi translater*", Apple-ISS technical report, 1997

[13] Richard Sproat, Chilin Shih, William Gale and Nancy Chang, "A Stochastic Finite-State Word-Segmentation Algorithm for Chinese", Computational Linguistics, Vol 22, Number 3, 1996

[14] Jian-Cheng Dai and Hsi-Jian Lee, "Paring with Tag Information in a probabilistic generalized LR parser", (1994), International Conference on Chinese Computing, Singapore

[15] K. R. Castleman, "Digital Image Processing", Prentice-Hall Signal Processing Series, Prentice-Hall Inc., USA, 1979.

[16] Wirote Aroonmanakun ,"Collocation and Thai Word Segmentation", In Proceedings of the Fifth Symposium on Natural Language Processing & The Fifth Oriental COCOSDA Workshop.

[17] Virach Sornlertlamvanich, Tanapong Potipiti And Thatsanee charoenporn, "Automatic Corpus-Based Thai Word Algorithm Extraction with the C4.5 Learning", Proceedings of the 18th conference on Computational linguistics, 2000.

[18] Thanaruk Theeramunkong and Sasiporn Usanavasin , "Non-Dictionary-Based Thai Word Segmentation Using Decision Trees ", Human Language Technology Conference, Proceedings of the first international conference on Human language technology research, 2001.

[19] Xin-Jing Wang,Wen Liu,Yong Qin ,"A Search-based Chinese Word Segmentation Method", International World Wide Web Conference, Proceedings of the 16th international conference on World Wide Web,2007

[20] Krisda Khankasikam and Nuttanart Muansuwan , "Thai Word Segmentation a Lexical Semantic Approach".

[21]Choochart Haruechaiyasak, Sarawoot Kongyoung and Matthew N. Dailey, "A Comparative Study on Thai Word Segmentation Approaches", 2008

[22] Pak-kwong Wong and Chorkin Chan, “Chinese Word Segmentation based on Maximum Matching and Word Binding Force”. In Proceedings of the 16th conference on Computational linguistics ,1996.

[23] Li Haizhou and Yuan Baosheng , “Chinese Word Segmentation”. In Proceedings of the 12th Pacific Asia Conference on Language, Information and Computation, PACLIC-12, 1998. 212-217.

[24]. Charoenpornsawat, P., Kijsirikul, B. 1998. “Feature-Based Thai Unknown Word Boundary Identification Using Winnow”. In Proceedings of the 1998 IEEE Asia-Pacific Conference on Circuits and Systems (APCCAS’98).

[25]. Meknavin. S., Charenpornsawat. P. and Kijsirikul. B. 1997. “Feature-based Thai Words Segmentation”. NLPRS, Incorporating SNLP.

[26]. Blum, A. 1997. “Empirical Support for Winnow and Weighted-Majority Algorithm: Results on a Calendar Scheduling Domain”, Machine Learning, 26:5-23.

[27]Daniel Jurafsky, James H. Martin. “Speech and Language Processing”.

[28] <http://www.crulp.org/oud/default.aspx>

[29] S.Hussain www.LICT4D.asia/Fonts/Nafees_Nastalique, in the Proceedings of 12th AMIC Annual Conference on E-Worlds: Governments, Business and Civil Society, Asian Media Information Center, Singapore, 2003

[30] Sarmad Hussain ,"Resources for Urdu Language Processing", In Proceedings of the Sixth Workshop on Asian Language Resources, 2008.

[31] <http://www.crulp.org/software/ling_resources/UrduNepaliEnglishParallelCorpus.htm>

[32] Sajjad, H."Statistical Part-of-Speech for Urdu", MS thesis , Centre for Research in Urdu Language Processing , National University of Computer and Emerging Sciencies , Lahore , Pakistan ,2007.

[33] Stanley F. Chen and Joshua T. Goodman, "An Empirical Study of Smoothing Techniques for Language Modeling", In Proceedings of the 34th Annual Meeting of the Association for

[34] MacKay, David J. C. and Linda C. Peto ," A hierarchical Dirichlet language model ", Natural Language Engineering, 1(3):1-19, 1995.

[35] Church, Kenneth W. and William A. Gale. 1991," A comparison of the enhanced Good-Turing and deleted estimation methods for estimating probabilities of English bigrams" , Computer Speech and Language, 5:19-54.